

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
1 March 2001 (01.03.2001)

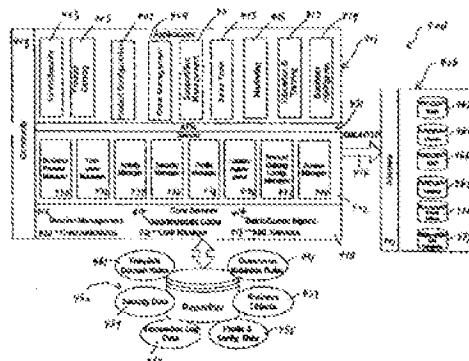
PCT

(10) International Publication Number
WO 01/14962 A1

- (51) International Patent Classification⁷: G06F 9/44
- (21) International Application Number: PCT/US00/22032
- (22) International Filing Date: 11 August 2000 (11.08.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
- | | | |
|------------|-------------------------------|----|
| 60/164,021 | 23 August 1999 (23.08.1999) | US |
| 09/440,326 | 15 November 1999 (15.11.1999) | US |
| 09/439,764 | 15 November 1999 (15.11.1999) | US |
| 09/547,161 | 11 April 2000 (11.04.2000) | US |
- (71) Applicant (for all designated States except US): ASERA, INC. [US/US]; 600 Clipper Drive Suite 100, Belmont, CA 94002 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): NORI, Anil, K. [US/US]; 5816 Newgate Court, San Jose, CA 95138 (US). BOONLEUNG TOMNU, John, K. [TH/US]; 205 Houghton Street, Mountain View, CA 94041 (US).
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KO, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR PROVIDING CUSTOM CONFIGURABLE BUSINESS APPLICATIONS FROM A STANDARDIZED SET OF COMPONENTS



WO 01/14962 A1

(57) Abstract: At least one method and apparatus for providing custom configurable business applications from a standardized set of components. A server device is configured to include best-of-breed components which comprise the server device. Best-of-breed server applications are also included which provide server functionalities. A modular set of components are used to construct a business service application according to user requirements. The components might include business steps which are operations with a defined set of input and output ports, with each port having a defined set of parameters, and business rules which are used to capture customer specific business practices. The business rules can also be used to produce different results depending on conditional triggers that are associated with the underlying business objects. Data representations between the server device and systems external to the device, as well as data representations between the users and the business service applications are utilized as business objects. A data repository is used for storing and retrieve data associated with the various business service applications and server functionalities. A community enablement feature allows data from many different sources to be aggregated, viewed and/or shared so that a user might conveniently analyze information relating to a particular product, wherein the retrieved information might be context sensitive.



Published:

.... *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR PROVIDING CUSTOM CONFIGURABLE BUSINESS APPLICATIONS FROM A STANDARDIZED SET OF COMPONENTS

RELATION TO PRIOR APPLICATIONS

5

This application claims priority of (1) U.S. Provisional patent application having serial number 60/164,021, filed August 23, 1999, entitled "Method and Apparatus to Provide Custom Configurable Business Applications From a Standardized Set of Components," and of (2) U.S. patent application (Attorney Docket No. ASRAP001) having serial number 09/440,326, filed November 15, 1999, entitled "Method for Providing Custom Configurable Business Applications From a Standardized Set of Components," and of (3) U.S. patent application (Attorney Docket No. ASRAP002) having serial number 09/439,764, filed November 15, 1999, entitled "Apparatus to Provide Custom Configurable Business Applications From a Standardized Set of Components," and of (4) U.S. patent application (Attorney Docket No. ASRAP004) having serial number 09/547,161, filed April 11, 2000, entitled "Business Methods For Providing Custom Configurable Business Applications From A Standardized Set Of Components," all of which is hereby incorporated by reference in their entirety.

FIELD OF THE INVENTION

20

The present invention relates generally to methods and apparatuses for providing custom configurable business or channel applications from a standardized set of components. More specifically, the present invention allows each business to select from a set of applications, customize that set of applications, and/or develop new customized applications from a set of development components.

25

BACKGROUND OF THE INVENTION

Business interactions have traditionally involved collections of people and/or other businesses that add up to form certain supply chains or demand chains. A supply chain is sometimes referred to as "upstream" from an enterprise business, with the manufacturers being on the "buying side" of the enterprise. A supply chain is comprised of various parties who facilitate bringing various materials from a supply source to the manufacturer. A demand chain is sometimes referred to as "downstream", with the customers being on the "selling side." A demand chain is generally comprised of the various parties who

facilitate bringing a product from an enterprise (or manufacturer) to a trading partner (or user). In order to complete the various transactions between these parties, person-to-person interactions have traditionally been used at the various points along the chain. Hardcopy forms and the like often accompanied these interactions. Today, the rapid development of 5 different electronic media to exchange information has led to many new approaches for the completion of certain business related tasks.

Computer networks provide an efficient means for transporting data between workstations or terminals on (or connected to) the network. Such networks can consist of Local Area Networks (LANs), which are generally restricted to one geographical area or 10 location. Such networks can also include Wide Area Networks (WANs) which connect a number of machines over a larger geographic area. The Internet is also an example of one such network. The Internet is a worldwide system of computer networks -- or a network of networks -- wherein users at any one computer can, if they have permission, get information from any other computer. The Internet was conceived by the Advanced 15 Research Projects Agency (ARPA) of the U.S. government in 1969 and was first known as the ARPANet. The original aim was to create a network that would allow users of a research computer at one university to be able to "talk to" research computers at other universities. A side benefit of the ARPANet design provided that messages can be routed or rerouted in more than one direction, the network can continue to function even if parts 20 of it were destroyed in the event of a military attack or other disaster (including simple down-time of component parts).

Today, the Internet is a public, cooperative, and self-sustaining facility accessible to hundreds of millions of people worldwide. The Internet is thereby providing a low-cost medium for the exchange of information between certain authorized parties. A large 25 number of merchants are using the Internet to directly offer goods and services to users or customers. Additionally, the Internet is being used as a transport medium for information between merchants and other related parties, particularly on the demand-side. A merchant might, for example, interact with certain Original Equipment Manufacturers (OEMs), distributors, direct retailers, and/or Value Added Resellers (VARs) through Internet links 30 back to the merchant's website. Such interactions would likely decrease the number of phone calls, Fax interactions, and/or person-to-person interactions, thereby saving time and money for both parties involved.

The rapid exchange of information over the Internet has led developers to automate certain business processes, including for instance, sales related tasks. Developers have previously devoted considerable attention to the automation of "back-end" tasks (e.g. the completion of forms, the order processing in the back-end ERP system, and the like).

- 5 Manufacturers already see cost savings in automating such back-end processes. On the demand-side, there are many business processes between the manufacturer and the buyers (trading partners) that can be automated. For instance, the process of determining how much of a discount should be offered for each individual buyer can be automated. Another task might include management activities regarding various business channels.
- 10 "Channels" are usually referred to as being direct, or indirect. An example of a direct channel is the company's sales force, which then interacts with the customers. An example of an indirect channel is a VAR, which might include a retail store, or the like. Still other tasks might include the customer support, storage and recall of marketing leads, and the administration of sales promotions. By automating more of such tasks, the sales
- 15 force for any given manufacturer can be made more efficient, thereby reducing the overall cost of selling. Automating business processes like selling, customer support, marketing/sales promotions etc. in a demand-chain can provide significant cost savings for manufacturers.

- 20 Certain new approaches have focused more on how to automate the full set of transactions relating to the channel relationship. This might include buying and selling, customer support, generation of new market leads, sales promotions, etc. Also, rather than have certain transactions take place over private enterprise networks (and the like), the newer approaches are attempting to automate all such processes over the Internet (often referred to as the "web"). Hence, all parties along the entire chain (e.g. customer, VAR, distributor, manufacturer) will be offered automated web-based services. Developers, however, have not implemented a complete and favorable demand-chain solution that automates, or allows ready creation of, customized applications throughout the downstream flows.

- 25 Automation of demand chain over the internet requires breadth of functionality to be integrated in one place. For example, the manufacturers need to provide functionality for buyers to authenticate themselves, browsing product catalogs, configuring products, submitting orders (to the back-end ERP systems), checking order statuses and so on. Manufacturers need to acquire a variety of products to provide for the demand-chain

functionality and integrate them in a coherent fashion. As the internet technology is proceeding at rapid rate, manufacturers are left with a daunting task of picking the best-of-breed products to provide the desired functionality and integrating them.

Internet transactions can also generally be divided into two categories: 1) business-to-business transactions, and 2) business-to-consumer transactions. Most solutions to automate transactions have dealt with business-to-consumer interactions. As such, these interactions are much more straightforward than business-to-business transactions. In a business-to-customer transaction, the merchant supplies a "storefront" or web site that offers products to any number of diversified consumers who might wish to view this web site. The consumer then purchases a product via a selection and payment method, and the product is thereafter shipped to the consumer. On the other hand, when one business deals with another business, there is a much greater amount of business processes and customization in the transactions that occur. For instance, the interactions that occur between a large petroleum manufacturer and a small chain of gasoline-service stations might be very different from the interactions that occur with a much larger (globally-based) chain of gasoline-service stations. Different volume discounts, incentive plans, and the like, might be offered to these different types of purchasers. Business to Business transactional systems lack facilities to empower users with necessary knowledge to run their day to day responsibilities. Manufacturers need a web based system to improve trust and loyalty, educate their customers, encourage peer to peer communication, make their users aware of the current industry trends: what the current system offers as Community services. There is a need to provide a seamless integration between the commerce services and the community services enabling manufactures to promote their products in an efficient manner. Such community features are much needed for rapidly growing vertically focused market makers called "Net Market Makers". These net market makers use the internet as their only medium to promote their products from different manufacturers.

Accordingly, certain in-house systems (i.e. developed internally) have been developed by manufacturers in order to handle business transactions relating to their particular product line (e.g. Cisco, which manufactures and sells routers). The in-house system is configured to handle the various interactions necessitated by different customers desiring different transactions. Such in-house systems, however, can prove to be very expensive to develop and maintain, and also may lead to integration of products rather

than an integrated product. A customized system might take years to develop, and require many hours of employee time to implement. Most small businesses do not have the necessary resources. Additionally, small businesses cannot justify the risk (or time) involved in developing such an in-house system, wherein the risk involves potential delays 5 in time-to-market with a particular product. As a result, such businesses must continue to operate under older models and practices, and therefore no savings can be realized from such automated services.

There exists a need for a system (i.e. method, or the like) that will address certain difficulties in customization of automated business services (across the Internet and other 10 networks), particularly relating to business-to-business transactions. The system should provide such services in a more timely and generally accessible manner. The system should provide for a more adaptable and scalable application solution across any number of products and/or technologies. A user should be able to utilize this system without incurring significant startup or development costs. Certain business applications should be 15 provided which can be customized according to each user's needs. Other business applications should be readily created using development routines and tools. The resulting applications should seamlessly interface with the overall system architecture. This approach should therefore provide rapid deployment of automated business applications, as customized for each individual company using the system.

20

SUMMARY OF THE INVENTION

To achieve the foregoing, and in accordance with the purpose of the present invention, the methods and apparatuses described include, for example purposes, a server device referred to as the Asera Commerce Server (ACS). The ACS provides for the development and implementation of customized, automated, web-based business service 25 applications. The ACS provides for a server based method wherein best-of-breed services and/or applications are integrated in a seamless fashion and offered to enterprise businesses which develop customized business service applications through using the system.

The ACS is designed for vendors that want to dramatically improve their indirect 30 selling efficiency, in addition to developing channel loyalty to capture a greater percentage of sales. ACS enables manufacturers to better manage their distribution channels and key relationships with channel partners -- i.e. distributors and dealers -- by optimizing high overhead channel management functions such as order entry, product configuration, order

status, promotional campaigns, customer service, technical support, and warranty/claims services. One resulting goal of the ACS is to provide an e-business platform for manufacturers and dealers, by providing transaction, marketing, pre sales, post sales, and community services.

5 ACS changes the nature of channel relationships by facilitating the development of fully automated transactions. Such transactions were previously conducted manually through implementations of self-service concepts. The ACS further provides the infrastructure for manufacturers to add significant value to the dealer network by providing community services that will allow dealers to interact with each other. An array
10 of customized community services, such as email, industry forums, and transaction services will serve to encourage and promote inventory exchanges and best-practices sharing -- thus contributing further to channel efficiency and channel loyalty.

Most manufacturer-dealer interactions today consist of phone, Fax, FedEx, and perhaps email. Automating these relationships substantially reduces the bottom line expenses. Automating these interactions also improves top line-selling efficiency because manufacturers make it easier for dealers to conduct business with them. If an enterprise were to start from scratch and develop similar features offered by ACS, the costs would be prohibitive, and the development effort would carry a long time-to-market penalty (or opportunity cost). Examples of in-house channel management systems have taken 4-5
15 years to develop, and have been known to carry annual budgets of \$50 million.
20

The ACS substantially reduces entry costs and eliminates the opportunity costs. The present system assembles best-of-breed components into a seamless outsourced service. The ACS provides improved management of the manufacturer-distributor relationship in high technology market sectors where indirect distribution channels add significant value
25 to the sales process. The net results of using ACS are improved competitive positioning for a company in the near term, and higher return on assets in the long term.

The ACS includes a Commerce Server that provides a core set of technology (or application) services. A unique architecture and framework are provided by the Commerce Server, and which facilitates the methods described herein. All interactions
30 with external systems or users are managed as business objects. The service application code is maintained separate from the data, thereby enabling the system to quickly include or change new business processes or technology components without having to write substantial amounts of new code. The business result is more rapid customer deployments

and/or modifications that are customized to the level of including (if desired) the proprietary or competitive business practices of a contracting company.

The ACS thus can be viewed as an ASP, or Application Service Provider. An ASP is generally an outsourcing business model. The ASP business model requires an open and extendable architecture that allows a system to implement a customer specific business solution in a short period of time. The ACS takes best-of-breed applications and incorporates them into one integrated solution to provide the ASPs. The architecture is scalable and extensible. A customized business (or channel) application solution is built for each enterprise company. The solution uses a "modular" or step-wise or "plug and play" approach towards building new applications. An enterprise company can then quickly acquire a turn-key e-commerce solution to automate their channel relationships. The present system presents little (or no) risk for the enterprise company because a solution is built by the present system. The costs of undertaking such a development exist as a fixed development cost of the present system. Any resulting customized solutions are implemented in considerably less time than previous systems. The enterprise company might pay for the application services on a cost per transaction, or a fixed fee basis.

The ACS is used to capture the particularized (or specific) business processes for a given customer, and these business processes are converted into a set of customized applications. The ACS uses business steps, and rules to construct the application. The objects are data representations. The steps are business operations with a defined set of input and output ports, with each port also having a defined set of parameters. The business rules are used to capture customer specific business practices. A unique tool that employs a graphical user interface (GUI), allows a developer to arrange various steps (or composite steps) into business processes, or workflows. The tool provides library catalogs of steps to be applied to the various objects. The connections between steps are also verified as correct. A graphical display of the business process is shown, and rules can thereafter be applied to provide further customization by conditionally tagging certain points. Hence, to create a business process (or application) for any given business, tools are provided which allow modules (or steps) to be plugged or dropped into the potential process. The steps can be moved, or the connections modified. An initial person-to-person (or other type of) interview with the business (or customer) can be used to produce the framework for arranging the steps according to the needs of that particular business (i.e. customized routines). The modular aspect of the present system allows this to be

done -- and modifications made -- in a relatively quick fashion. For instance, if a process has been created, but the customer wants it to behave in two different manners, then certain rules can be applied to provide the desired results, depending on conditional triggers that can be associated with the underlying business objects.

- 5 The underlying commerce server provides certain functional components. Thereafter, various vertical applications are built (and/or customized) for the particular enterprise business. A suite of best-of-breed components and underlying applications are chosen to comprise, or run on top of, the commerce server. These components can be selected as the best technologies to provide certain desired efficiencies, cost reductions,
10 and the like. The best-of-breed components are thereafter incorporated (or interfaced) with the business applications through APIs. A catalog of APIs can be provided for developers of applications. If any underlying best-of-breed components change, then the commerce server configuration can be changed without interfering with any user interfaces.
- 15 Community aspects are also offered between the various developed applications, and server functions. Community features allow access to data and information from different functional components of ACS and other external data, thereby providing more information to the channel participants. Users can generally jump back and forth between data on the system, without having to connect to another server or site. Usage of such data
20 might also be made context sensitive to speed through various tasks, and eliminate unwanted information. For instance, a person might be in an application that would lead to the purchase of security products. A context sensitive ability would switch the user to a community area in order to see industry reviews, magazine reviews, competitor comments, and/or latest trends, etc. about such security products. Examples of community
25 features might include message boards, live chat links, new links, new product promotions, research links, etc.

The present system also offers features wherein the user can request for an aggregation and presentation of different data or information into one convenient display area. The present system employs a template management component to facilitate such features. A master-template might contain enterprise information (i.e. logos or color schemes) that should be applied across a series of pages. A micro-template, or a collection of micro-templates as a wireframe, is used to selectively display dynamic data from a variety of sources. The data is dynamically linked into the page from a designated data

source, and displayed on the appropriate part of the page. The data can therefore be displayed or aggregated in any manner specified by a developer of the page, or by the application displaying the page.

The architecture of the present system thereby provides, among other things, a commerce server, an application development tool, and a community enabler. The architecture is extensible and scalable. A user of the system can acquire a turn-key set of customized business applications. The solution provides a high degree of reliability because the applications are developed and implemented by the centralized ACS. Single-source log-on is also provided, as the user does not have to move from server to server.

Information can thereafter be shared between various information sources via the community feature.

The business features of the present system include the same turn-key operability and community features. The business enterprise is provided with a low cost and low risk entry into automation of various business practices. The business practices of an enterprise can be captured and developed into a customized set of business processes. Best-of-breed components are offered to run the business applications on the underlying commerce server. A business (or customer) can improve its channel effectiveness with a quick roll-out of such services. Through hosting of various businesses on one server, aspects of the community features are enabled and enhanced.

Accordingly, one aspect of the present invention includes at least a method for providing custom configurable service applications from a standardized set of components associated with a server device, the method comprising: configuring best-of-breed components and associated server device applications for communicating with systems external to the server device; forming and utilizing a first set of objects which are data representations for interacting with the systems external to the server device; constructing a service application from a modular set of application steps and application rules to perform services according to user requirements, the service application interfacing with the best-of-breed components; and forming and utilizing at least a second set of objects which are data representations for interacting with a user through the service application.

Still another aspect of the present invention includes at least a method for providing custom configurable business service applications to a user through a server device, the method comprising: configuring the server device as comprised of best-of-breed components; configuring the server device with best-of-breed server applications for

interacting with the best-of-breed components and providing server functionalities; constructing a business service application from a modular set of components that can be dynamically arranged according to user requirements, the business service application interacting with the server applications to provide the user with the server functionalities; 5 and interfacing with systems external to the server device via adapters configured to communicate with the external system.

Another aspect of the present invention includes at least an apparatus for providing custom configurable service applications from a standardized set of components associated with a server device, the apparatus comprising: a server device configured 10 from best-of-breed components, and running best-of-breed server applications to provide certain server functionalities, the server device having server data including data representations for interacting with systems external to the server device, the data representations being managed as objects; a graphical design tool used for constructing at least one service application from a modular set of application steps and application rules, 15 the service application performing services according to user requirements, the service application being capable of interfacing with the best-of-breed components and server applications, the server data further including data representations for interacting with users, the data representations being managed as objects; at least one adapter device for interfacing with the systems external to the server device; and a repository for storing and 20 retrieving server data, the repository facilitating centralized storage, retrieval, and sharing of server data between all the server device components, server applications, and service applications.

Still another aspect of the present invention includes at least an apparatus for providing custom configurable business service applications to a user through a server 25 device, the apparatus comprising: a server device as comprised of best-of-breed components; the server device configured with best-of-breed server applications for interacting with the best-of-breed components and providing server functionalities; a construction tool for developing a business service application from a modular set of components that can be arranged according to user requirements, the business service 30 application interacting with the server applications to provide the user with the server functionalities; and an interface between the server device and systems external to the server device which utilizes adapters configured to communicate with the external systems.

Another aspect of the present invention includes a method for providing customized business application solutions to at least one enterprise company that contacts a centralized provider of such solutions, the method comprising: assembling a collection of best-of-breed components to provide a seamless connection to services that are 5 outsourced from the centralized provider; capturing the particularized business processes of the at least one enterprise company; providing a modular approach towards building a business service application for the at least one enterprise company; and customizing the modularly built business service application according to the captured business processes.

Still another aspect of the present invention includes a business method for 10 providing a user with a centralized framework for developing and customizing a business service application from a modular set of components associated with a centralized device, the method comprising: configuring the centralized device with best-of-breed components for communicating with sources external to the centralized device; capturing the particularized business processes of a user; providing certain modular components that can 15 be arranged to form a business service application; and customizing the arrangement of the modular components to form a business service application according to the particularized business processes of the user.

Still another aspect of the present invention includes A business method for providing a plurality of standardized business service applications and customized 20 business service applications through a central server device, the method including: configuring the server device from best-of-breed components; configuring the best-of-breed components to communicate with data sources external to the server device via application interfaces (APIs); determining the business processes of a user business that contacts the central service device; forming a customized business service application 25 from existing standardized business service applications as modified by modular components provided by the central server device; and providing for community interaction between business service applications, and the central server device.

These and other aspects and advantages of the present invention will become apparent upon reading the following detailed descriptions and studying the various figures 30 of the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention, together with further advantages thereof, may best be understood by reference to the following description taken in conjunction with the accompanying drawings in which:

5 FIG. 1 is a prior art block diagram of certain representative elements for an extended value chain.

FIG. 2 is a prior art plot showing the trends in IT spending as enterprise businesses move towards automating customer relationships.

10 FIG. 3 is a block diagram, according to one aspect of the present invention, which illustrates a hosting architecture.

FIG. 4 is a block diagram, according to one aspect of the present invention, which illustrates the ACS architecture.

FIG. 4A is a block diagram, according to one aspect of the present invention, which illustrates the back-button feature.

15 FIG. 4B illustrates, according to one aspect of the present invention, an example of the overall Asera portal screen.

FIG. 4C is a block diagram, according to one aspect of the present invention, which illustrates certain vertical application and community features in relation to the server.

20 FIGS. 4D-1 and 4D-2 illustrate, according to one aspect of the present invention, an example of community feature screens.

FIG. 5 is a block diagram, according to one aspect of the present invention, which illustrates content aggregation of information on a displayed page.

FIG. 6 is a block diagram, according to one aspect of the present invention, which illustrates a catalog application, and its interaction with other applications in the system.

25 FIGS. 6A-6D illustrate, according to one aspect of the present invention, certain screens associated with the product catalog configuration manager.

FIG. 7 is a block diagram, according to one aspect of the present invention, which illustrates an object data flow.

FIG. 8 is a block diagram, according to one aspect of the present invention, which illustrates an example business object flow.

5 FIG. 9 is a block diagram, according to one aspect of the present invention, which illustrates an example business process flow.

FIG. 10 is a block diagram, according to one aspect of the present invention, which illustrates a template arrangement.

10 FIG. 11 is a block diagram, according to one aspect of the present invention, which illustrates an example composite step.

FIG. 12 is a block diagram, according to one aspect of the present invention, which illustrates an example interactive step.

FIG. 13 is a block diagram, according to one aspect of the present invention, which illustrates an example application designer window anatomy.

15 FIGS. 13A-13O illustrate, according to one aspect of the present invention, certain example screens associated with the graphical business application design tool.

DETAILED DESCRIPTION OF THE INVENTION

The present invention offers a low risk alternative for companies that wish to
20 quickly deploy an automated business application to their customers. According to the traditional model, a company must either develop (or have developed) an in-home solution or evaluate other products to be integrated. Thereafter, personnel within the enterprise company must be trained on the usage of such software, and complete its evaluation. A prototype of the software product is thereafter developed based upon feedback from the
25 evaluation process. When the software is completed, a beta-deployment of the associated application can thereafter made to the public. This lengthy process might cost upwards of \$2 million, and take the developers 8-12 months to complete, depending on the complexity of the application program. The success of such a program in generating (or saving) revenues is still unknown.

In contrast, the present invention offers methods and apparatuses for providing a business application that has been developed and is customizable for the particular business enterprise. For instance, the present system is capable of generating -- within approximately 60 to 90 days -- a working business application solution (or set of solutions) for the business enterprise. The business enterprise does not pay large sums of money for development costs. Instead, the business enterprise might elect a plan that includes paying the administrator of the system a certain amount for each transaction (i.e. a fixed sum, usage fee, or a percentage of the value of the transaction).

The ACS (Asera Commerce Server) includes server infrastructure, tools, and integration with other products that provides for the development and implementation of customized, automated, web-based business service applications. The ACS provides for a server based method of implementation wherein best-of-breed products, applications and/or services are integrated in a seamless fashion and offered to enterprise businesses which develop customized business service applications through using the system.

The ACS is used to capture the particularized, or specific business processes for a customer, and that business process is converted into a set of customized applications. The ACS uses business objects, steps, and rules to construct the application. The objects are data representations. The steps are objects with a defined set of input and output ports, with each port also having a defined set of parameters. The business rules are used to further customize the interactions between the various business objects. A unique tool that provides a graphical user interface (GUI) allows a developer to arrange various steps (or composite steps) into business processes, or workflows. The tool provides library catalogs of steps to be applied to the various objects. The connections between steps are also verified as correct. A graphical display of the business process is shown, and rules can thereafter be applied to provide further customization by conditionally tagging certain points. Hence, to create a business process (or application) for any given business, tools are provided which allow modules (or steps) to be plugged or dropped into the potential process. The steps can be moved, or the connections modified. An initial person-to-person (or other type of) interview with the business (or customer) can be used to produce the framework for arranging the steps according to the needs of that particular business (i.e. customized routines). The module aspect of the present system allows this to be done -- and modifications made -- in a relatively quick fashion. For instance, if a process has been created, but the business (or customer) wants it to behave in two different manners,

then certain rules can be applied to provide the desired results, depending on conditional triggers that can be associated with the underlying business objects.

The underlying commerce server provides certain functional components. Thereafter, various vertical applications are built (and/or customized) for the particular enterprise business. A suite of best-of-breed components and underlying applications are chosen to comprise, or run on top of, the commerce server. These components can be selected as the best technologies to provide certain desired efficiencies, cost reductions, and the like. The best-of-breed components are thereafter incorporated (or interfaced with) the business applications through APIs. A catalog of APIs can be provided for developers of applications. If any underlying best-of-breed components change, then the commerce server configuration can be changed without interfering with any user interfaces, or other ACS applications.

Community aspects are also offered between the various developed applications, and server functions. Community features allow access to data and information from different functional components of ACS and other external data, thereby providing more information to the channel participants.

Referring to Figure 1, a prior art block diagram 100 is shown of these various value chain relationships. An enterprise business 102 is shown at the center of the chain. A demand chain 104 extends up from the enterprise 102 to the customers 106. The demand chain might include the illustrated example parties of an OEM (outside equipment manufacturer) 108, distributor 110, direct retailer 112, or VAR (value added reseller) 114. A supply chain 120 is shown with a series of manufacturers 122, 124, and 126 providing goods/services to the enterprise business 102.

Referring to Figure 2, a prior art plot 200 is shown of a typical enterprise IT agenda. This chart reveals that as the move is made more towards automating customer relationships, the IT spending for the enterprise will reach levels which most companies cannot sustain. This is true even if there is a large value returned for such development efforts. Hence, the present invention offers automated customer solutions that have been developed, and are maintained, external to the customer. This allows the customer to acquire these value added services, but without the IT expense.

HOSTING ARCHITECTURE. Referring now to Figure 3, a block diagram 300 is shown of certain representative elements used to host the architecture of the present

system. A set of dealers 302, are shown using browsers (or the like) 304 to interact with the Internet (or other similar network) 306. The hosted service 308 of the present invention includes at least one router 310 that interacts through firewalls 312 with redundant authentication servers 314. These servers 314 are then in communication with redundant Asera Commerce Servers (ACS). The servers would use co-location services such as UUNet, Exodus, or Qwest to transport data between devices. A replicated data center 318 is shown for redundancy purposes. An administrator 320 interacts with the hosted service 308. The administrator would provide maintenance of the system, and/or human interaction with the enterprise clients using the system. A manufacturer network 322 is shown including a router 324, firewall 326, and computer 328. The manufacturer network 322 might interact with the hosted service 308 via a dedicated connection 330. The dealers 302 might interact with the hosted service 308 via a connection 332 through the internet 306. A virtual private network (VPN) 334 can also be used to interact with the various elements shown. A VPN is a private data network that makes use of the public telecommunication infrastructure, maintaining privacy through the use of tunneling protocol and security procedures. A virtual private network can be contrasted with a system of owned or leased lines that can only be used by one company.

ASERA COMMERCE SERVER ARCHITECTURE. Referring now to Figure 4, a block diagram 400 is shown of certain representative elements that might be used to form the Asera Commerce Server (ACS) configuration 401 of the present invention. The present invention includes a configuration comprised of certain component parts including: 1) A server device 402 upon which the various applications run and interact. This server is sometimes referred to as the Asera Commerce Server, or application server; 2) Vertical applications 404 that interact with the outside users; 3) a back-end (ERP, or otherwise) interface 406 to communicate with various back-end systems; and 4) a community interface environment 408 wherein information can be a shared aggregation from many different applications, and many different information source areas. The application server uses certain best-of-breed components, and makes use of standard interfaces to utilize the server framework. A server might then be built, using off-the-shelf components or otherwise, which can run multiple applications on the server. The applications can be customized (or developed particularly) for the enterprise business user, as described in further detail below.

Still another feature of the present system includes scalability. Known systems operating in a web environment cannot generally scale their architecture down to a desktop environment. To achieve full functionality (and scalability), virtually any full blown applications server might be used. However, a party might also choose to use a 5 personalized server, which they can run on desktop or remote machine, in a disconnected mode without any networking. In such a mode, the present invention can still be implemented using virtually the same architecture. In the latter case, a "light-weight" web-server engine can be used that will provide the same system, but one that will run on a smaller device, i.e. a laptop or PC. Any level of scaling might be achieved between 10 these two example versions. If a desktop (or non-network) version is used, only certain best-of-breed components might be incorporated so as not to overload the system. Such components are identified and used on the desktop, with synchronization provided for the various back-end connections needed.

15 APPLICATIONS. A set of "vertical" applications 404 are shown illustrated for the present example. One important aspect of the present invention is that any application might be developed and offered in this area, through the development of a business process, or workflow. The workflow uses business objects, steps, and rules (as further described below). Moreover, a certain set of applications might be developed and offered to an enterprise user, with customization being performed on the applications to suit each 20 particular user. The applications can be grouped based on their functions, such as commerce community, pre-sales, support and services. The illustrated applications include administration/security 403, product catalog 405, product configuration 407, order management 409, support/service management 411, sales tools 413, marketing 415, education and training 417, and business intelligence 419.

25 The applications might be developed as stand alone products within the present system. Preferably, applications will be developed to integrate with the overall system, and thereby facilitate the flow of data/information between the various applications. A user might enter one application after another, with the sequence comprising a meaningful transaction not readily offered by other systems.

30 CORE SERVICES. Certain core services 410 are provided which are necessary for the operation of the components and applications running on the server. Any of a variety of core services might be provided, and the following example core services are not meant to be limiting in any way. One core service would include session management

capabilities 412 for handling enterprise (or other types of) users that might contact the server device configuration 401. The session management service would take into account the general capabilities of the server to handle incoming session traffic, and allocating resources accordingly.

5 Another service would include data/metadata caching 414. Such caching would provide enhanced performance and connectivity. For instance, if an application were to request data from a back-end system, that application could continue to perform (while the cache had information to provide) without further requesting more information from the back-end. Additionally, if the flow of information were to lag, then the application might 10 still perform based on the various cached data. The back-end system will generally reside, for instance, behind the firewalls of various manufacturers (and the like), and a constant data flow is often beyond the immediate control of the present system. Continued operation might be achieved if throughput rates start to fluctuate, or if any data was misdirected or lost.

15 Yet another core service includes batch/queue management 416. Certain information might be queued (or batched) for forwarding to another device at a later time. Additionally, non-time critical information can be queued to provide more bandwidth for higher priority traffic. The system might thereby continue to operate in an efficient manner if a corresponding device were to cease functioning, or receiving data.

20 Still another core service includes XML services 418. A core language is generally needed in order to communicate with the various back-end devices. XML (Extensible Markup Language) is a flexible way to create common information formats and share both the format and the data over the World Wide Web, intranets, or elsewhere. For example, computer makers might agree on a standard or common way to describe the 25 information about a computer product (processor speed, memory size, and so forth) and then describe the product information using XML. Such a standard way of describing data enables a user to send an intelligent agent (i.e. a program) to each computer maker's Web site. Such an agent might gather data, and then make a comparison. Accordingly, XML can be used by any individual or group of individuals or companies that might want to 30 share information in a consistent way. XML is similar to the common web language HTML (Hypertext Markup Language). Both XML and HTML contain markup symbols used to describe the contents of a page or file. HTML describes the content of a Web page (mainly text and graphic images) in terms of its display and interaction. XML describes

the content in terms of what data is being described. XML is "extensible" because, unlike HTML, the markup symbols are unlimited and self-defining. XML is generally a simpler and easier-to-use subset of the Standard Generalized Markup Language (SGML) -- i.e. the standard pertaining to how to create a document structure. HTML (Hypertext Markup Language) and XML might both be used together in web applications.

In the present application, XML is employed as an emerging standard with certain advantages over HTML (however, HTML and other such languages might also be used). XML documents are gathered from the different best-of-breed products, and these XML documents are aggregated into displayable (and usable) pages on a web screen. The web pages are formatted in ways that conform to the needs and desires of the customer for certain pieces of information, with the XML information displayed accordingly. Formation of the web pages is aided by the use of templates (i.e. micro and macro templates), as further described below.

Still other core services include communications capabilities 420. A communications service would manage the flow of information from the various data sources connected to the server 401.

FUNCTIONAL COMPONENTS. Certain functional components also comprise the server 402. While not intended to be limited to such processes, the example configuration shown in Figure 4 utilizes certain functional components including: Business Process Manager 430, Template Manager 432, Activity Manager 434, Security Manager 436, Profile Manager 438, Content Aggregator 440, Product Catalog Configuration Manager 442, and Access Manager 444. Such components are individually described below.

BUSINESS PROCESS MANAGER. One of the main components is the Business Process Manager 430. This functional component is used to capture the interactions between a manufacturer and their channels. For instance, an enterprise business might deal with many different channels, or channel partners (i.e. a small business, a large business, a VAR, etc.). Each of these interactions will be different, and it becomes important to be able to customize such interactions between the channel partner and the enterprise business. The identity (or status) of the particular enterprise business might factor into the allowed interactions. Other factors might include the credit limit of the enterprise business, or the rated status of the enterprise (i.e. elite gold partner, regular

partner, etc.). All of these example factors need to be captured into the interactive business experience supplied between the enterprise and the channel partner.

This functionality is accomplished using a business flow engine. In general, a series of business steps are provided, wherein a client/user can utilize, or assemble, such 5 steps to build a desired business flow. The steps operate generally on objects. For example, if something is being sold, then an associated object might include the actual order being placed. Other such objects might include the product, the customer, and the shipping destination. Thereafter, a set of operations might be performed on the objects. For instance, an operation might check if the particular user is a valid user. Business steps 10 (simple, interactive, or composite) are defined that perform certain operations on objects in association with a set of defined input parameters and output parameters (or ports). A simple step is a basic unit. A composite step results from "stitching together" simple steps to form a larger composite unit. An interactive step involves an input or output from the application to an external interactive system. Ultimately, the steps (simple, interactive, or 15 composite) are interactively strung together (using the application design tool, or the like) to form a business application.

A set of business rules is also defined, and can be applied to the service application in order to further customize the flow of data. For instance, a rule might limit a particular customer to placing orders only between certain hours of the day. As such, the rules might 20 thereby be used to turn on (or off) the flow of data between certain points in the business process, depending upon a conditional input (which is the rule). When a collection of such rules have been applied, this service application has been custom built to perform a given set of operations on a fixed set of data associated with certain objects.

Further details regarding the development of the business processes are provided 25 with reference to Figures 8, and 9. A user tool is also provided to help build such applications, and is further described below with reference to Figure 13.

TEMPLATE MANAGER. Functional component 432 shows a template manager that is used to manage the display of data to the end user, on a web page or otherwise. Templates can provide "look and feel" and positioning customization of the pages which 30 are ultimately displayed to a user. Certain concerns addressed by such templates might include, for instance, the placement of an enterprise logo, or the appearance of certain screen sidebars, or the choice of colors to be associated with various parts of the screen. The display screens might also be customized depending upon what particular customer(s)

are going to use the screens. Hence, a series of templates is used to provide the visual display for various parts of the screen. Constantly re-occurring areas on the page, such as a business enterprise logo, or color scheme, would be covered by one such template. Such broad (and re-occurring) templates are often referred to as master-templates. Other 5 templates are used to provide the visual display for areas that dynamically vary according to certain data inputs. Notably, the data inputs might come from many different sources (i.e. graphical pictures from one area, statistical data from another area). Templates of this nature are referred to as micro-templates. A collection of micro-templates grouped to make a meaningful application is called a wireframe. A master-template in combination 10 with certain wireframes can form a macro-template. By using a selection of macro-templates, in combination with a selection of wireframes, a web display screen can be customized -- in a relatively short period of time -- that has the look-and-feel and positioning of a particular enterprise business that commissioned the custom creation of such display screens. With such customization, the screens provide an aggregation of 15 content from various sources, with the information on the page being arranged according to the needs of a user (or business enterprise serving the user).

Such templates are further described in relation to Figure 500, which shows a block diagram of certain representative template areas. A page 502 is shown that includes a macro-template for showing standard information 504. Yet another template 506 might 20 be used to show various user options associated with that page. Template 508 would be used as a content aggregation area for showing various micro-templates. Configuration area 510 would link to database 512 to provide configuration data. Area 514 might be used to display pictures of the product, as linked from database 516. Reviews of the product might be displayed in area 518, wherein the information is provided via a link to 25 database 520. The content aggregation area can be configured to include any data desired by a user. The data is thereby arranged for display via the use of the templates, through the template manager.

ACTIVITY MANAGER. The Activity manager 434 manages all incoming requests into the commerce server, and makes sure a user is guided in the appropriate 30 direction. Each object has an associated API. Developers use the various APIs in the formation of applications. A corresponding catalog of APIs is provided for developers to use in linking together various objects, and communicating with the ACS. A layer of APIs 431 therefore exists between the set of developed applications 404 and the functional

server components. The customizable aspect of each developed application is therefore made available to the developers through the use of this known set of APIs.

One example of activity manager functionality is to support the "back button" feature. The present invention prevents a user from getting lost or mis-directed if they use the "back button" on their browser in a transaction oriented web site. The various business flows of information are created and displayed to a user, but generally do not allow a user to "back up" through the process flow of such information. Prior solutions have used an error message that informs the user that they are not allowed to "back up." The present invention instead uses the invocation of the "back button" to direct the user to a proper page. Such direction might be context based. For instance, the direction might be contextually related to the task being performed by the user. Similarly, the direction might relate to the current page the user happens to be displaying, and so forth.

Referring now to Figure 4A, a progression 1500 is shown by a user through a series of pages. These example pages include Page 1 (view catalog) 1502, Page 2 (Order an item) 1504, Page 3 (Apply discount) 1506, Page 4 (view quote) 1508, and Page 5 (Add shipping information) 1510. If a user invokes the back-button on the browser, as shown by 1512, then the user might erroneously be directed from Page 4 back to Page 3. If these were done multiple times, then a discount might be improperly applied many times. The present solution instead directs the user back to Page 4, and an error message is displayed to instruct that user that he cannot return to Page 3.

SECURITY/ENTITLEMENT MANAGER. Still another functional component includes the security (or entitlement) manager 436. Traditionally, various systems being aggregated under a single configuration required an individual log-on (using passwords and/or certificates) for each system. This proves to be cumbersome and difficult to administer. Under the present system, a common sign-on (i.e. single sign-on, or log-on) is provided, and common unified security is offered for all systems accessible through the configuration. Moreover, once a user is authenticated, the system can entitle a user to be able to perform (or not) certain tasks. This is done through a set of defined security rules (or entitlement/authorization rules) which are applied to the objects/operations in the developed workflow. For instance, the rule might conditionally check if the user belongs to a certain organization. If so, that person might thereafter be allowed to place certain kinds of orders or allowed to receive certain discounts.

As for the authentication processes, no new products or software necessarily need to be created. Instead, best-of-breed products are preferably used to perform the authentication tasks. In general, the business process manager is thereby used to capture certain best-of-breed security measures through the application of entitlement or 5 authorization rules to the formed workflows. Once a single sign-on is achieved, then the user can interact with any of the authorized applications, which then interact through the back-end with any of the attached external (ERP, or other type) devices.

In the present invention, the security model is also integrated with the overall organizational model. Traditionally, security models are oriented towards protecting one 10 particular item in a model. The present system integrates well with such things as customer hierarchies and the like, including the various manager components and how they interact with each other.

For instance, a sales representative might use the single sign-on ability to access the system. That sales representative might be responsible for (and have access to) a 15 particular geographical area. Such geographical information might be accessed through an ERP device in one database area. The security information might be kept in yet another database. In the present system, all these components are brought together and integrated in a seamless fashion.

Also, through the use of the best-of-breed security products, most up-to-date 20 security functionality is made available to the manufacturers.

PRODUCT CATALOG CONFIGURATION MANAGER. Another functional component offered by the present invention includes the product catalog configuration manager 442. Various products carry with them the aspect of being configurable to a variety of different physical arrangements. An enterprise business is faced with the 25 prospect of trying to offer all of its different product descriptions and configurations to the end buyer. As yet another aggregate capability, the present device offers the ability to catalog -- and provide configuration choices -- for a enterprise product line. The configuration manager device is preferably a best-of-breed component that is integrated into the overall system.

30 An infrastructure is included that allows an administrator to create the product information. For instance, the identity of the product, the associated enterprise, the product description, and the like are retrieved from a back-end ERP system (or database).

This information is made available by the present server system so that the various business applications can readily access this product catalog-type information. The present system thereby serves as a staging area for such data, wherein it is much more efficient to access such data from the present system than through the back-end ERP connections. This arrangement enhances the availability of such data, and relatedly enhances the performance of the overall system.

In general, more sophisticated products have more complex configuration requirements. In order to complete this configuration task, access to certain product information is generally needed. Such information might include specification data, pricing information, discount information, and the like. The present system maintains a certain level of flexibility in presentation of such configuration data and information, so that different products lines and component parts can readily be accommodated.

This information is therefore readily available to complete the proper configuration of the product. The product catalog information serves as an anchor point for all the other applications to refer to such product information. Hence, the catalog manager serves, in many respects, like a product information database. In past configurations, product catalog information was simply used as that -- a product catalog. The present system, however, makes the information available for all the other applications, to use as anchor points.

Referring now to Figure 6, a block diagram 600 is shown of certain representative elements that comprise (and are associated with) the catalog configuration manager. A central catalog information database 602 is shown. This database 602 interacts with any of a variety of other databases, including for instance a marketing database 604, or a technical support database 606, or other ERP systems 608. A cataloging application 610 refers to the database 602 to form displays 612. A configuration operation 614 might thereafter be performed (or selected) using product configuration information from the catalog information database 602. The configuration will likely be associated with an identifiable name 618, such as "3Com Ethernet 500 combo", or the like. An order entry operation 616 might thereafter be performed to produce an order 620 that interacts with information for the database 602, and the name 618.

When switching back and forth between the database 602, and generalized elements that interact with the database 602, there are differences involved because of diversified product integration and the like. The present system thereby employs

alternative IDs 622 (or identifiers) as a field within the catalog information database 602. The alternative ID is used to link some (or all) of the information associated with the ID to the requesting element. An alternative Id for the above mentioned item might include "3C500combo." In general, the configuring device (or other such device) may not be able to readily access the catalog information in the database. However, in a functional way, the present implementation provides such access, with the alternative ID being the key (or index) into the catalog information database for that information, for configuration or other service.

Therefore, a user might view a particular page, and select line items in order to 10 configure a desired product. The line items serve as anchor points back into the catalog information database to retrieve the desired information (via use of the alternative ID). As described above, other functional components might also tap into the catalog information database via the alternative IDs.

Referring now to Figures 6A-6D, screen shots are shown of example pages relating 15 to the Product Catalog Configuration Manager, including the following: a catalog search page (Figure 6A), a product configuration page (Figure 6B), and quote pages (Figures 6C-6D).

ADAPTOR NETWORK TO BACK-END SYSTEMS. A set of back-end systems or databases are shown as element 406. Such systems or databases might include the 20 shown examples of a: Baan database 462, Oracle database 464, SAP 466, and Legacy database 468. Additionally, it might be desired to send a message to another type of system through an interface such as Front Office 470 (i.e. Siebel, Vantive), an EDI (Electronic Data Interchange) Gateway 472. In the present invention, the adaptors are capable of taking the business object information -- as related to the business steps 25 forming the various business application(s) -- and interfacing it with the various back-end systems. The business object information is transferred to the adapters 407 via link 474. The preferred language medium includes XML and/or HTTP. Other languages might similarly be used.

As mentioned above, any given application is interacting with business objects -- 30 as per the object oriented framework of the present invention. These business objects are used to capture the business data of an enterprise. Such business objects include order objects, user objects, customer objects (and the like) and operations are performed on these objects. An object abstraction is thereby written, and then this object abstraction is

mapped into the back-end ERP (or other type) systems. The integration of such back-end systems, database systems, vertical applications, and community features is one benefit provided by the present invention.

ACCESS MANAGER. Yet another functional component includes the access manager 444. This component is used to connect all the channels (or data interaction devices) within the server configuration 401 to external data sources, such as databases or other hardware systems. While not limited to such, the external hardware would typically include ERP devices, or the like. ERP (enterprise resource planning) is an industry term for the broad set of activities supported by multi-module application software that helps a manufacturer or other business manage the important parts of its business, including product planning, parts purchasing, maintaining inventories, interacting with manufacturers, providing customer service, and tracking orders. ERP can also include application modules for the finance and human resources aspects of a business. Typically, an ERP system uses, or is integrated with, a relational database system. The deployment of an ERP system can traditionally involve considerable business process analyses, employee re-training, and new work procedures. In the present system, the object (i.e. an order) that is defined in the server configuration 401 must be transformed into the back-end ERP system 406. Adapters 407 are provided between the access manager 444 and the ERP devices 406. According to one aspect, an Active Software adapter serves as an intermediary between the Access Manager and the ERP systems. Integration is also provided with EAI (Enterprise Application Integration) vendor products.

Referring now to Figure 7, a block diagram 700 is shown of certain representative elements of an object data flow. An example application 702 is shown sending a business object 704, which in turn might be an order object 706. A feature of the present invention is to provide ERP access via a foreign source adaptor, or the like. The ERP systems are integrated with the various databases, using the aggregate business environment. In this present example, the order object 706 is shown being divided into a first object part 708 and a second object part 710. The object parts can then be mapped onto different back-end ERP databases, or the like. In the present example, the first object part is shown being mapped to a first database 712, and the second object part is shown being mapped to a second database 714. Such "plug and play" modularity allows for the data aggregation and implementation of applications to communicate with a variety of foreign sources (ERP or otherwise).

COMMUNITY INTERFACE. The community interfacing feature allows the various channels to work more efficiently and in-sync with each other. In order for a channel to be successful, it becomes important for the channel to understand (and/or have access to) a variety of information that might affect business transactions. This is particularly true for channels that work in groups. In particular, the Internet has facilitated more community interaction than ever before. Through the use of this medium, channels might share information through chat rooms and such, in the context of certain transactional areas. Such sharing of information might be done through message boards, chat rooms, discussion forums, auctions, and the like. Accordingly, the ACS can provide an aggregate of desired information to the user on one page or area (or a series of pages, as desired).

Additionally, the community feature can be oriented towards various vertical types of operations. For instance, a group of enterprises might all be involved in a particular area such as chemicals. Business transactions relating to the particular chemicals will tend to have similarities. Such chemical producers might also tend to publish information pertaining to certain chemicals, including product reviews, analyses, criticisms, etc. The channel effectiveness is thereby increased by providing (and facilitating) the flow of information between the various channel participants. The community interface can enhance a variety of features, including the following:

Trust and Loyalty. The community feature is intended to improve trust and loyalty among the channel participants. Community aspects to facilitate such improvements would include the fact that users of the system are providing their own content to the pool of channel information. Trust and loyalty are further encouraged by making a channel participant more successful. One way to provide more success is by empowering the channel participant with more information.

Peer-to-Peer Communication. The community feature is also intended to encourage education and peer-to-peer communication. If a merchant has a product that is superior to other products, and a major industry publication is touting this fact, then the channel participant might want to make sure that this information is known. Research feeds and message boards can be used to facilitate such exchanges.

Industry Awareness. Expert advice and industry awareness might be facilitated through news feeds, or the like. Generally, any one industry includes a "net market maker" that collects information regarding that particular industry and provides an

objective evaluation document or reference (i.e. magazine or website). Information sharing would be facilitated (in general) by the complimentary services offered by the present invention.

Research. The present centralized system would greatly enhance research capabilities through community links. For instance, if a party had some interest in security or firewalls, a search on a common portal website would result in a large amount of "unfiltered" information. However, there are many existing (best-of-breed) companies that filter such content and can supply such information to the present system. It is also important that a researching party be subjected to unbiased, objective information regarding a product in order to build trust and loyalty. Because a variety of sources regarding a particular product can be readily accessed with the community feature, a product merchant is less able to tout the virtues of their product in an advertising type manner. Instead, objective messages from industry sources (and the like) are provided. This further builds trust and loyalty for the system from the buyer (and enterprise merchant) of such a product.

Product Launches. A new product could be launched and information offered -- via the community aspect -- to the many different companies tied into the ACS site.

Personalization and customization. User customization would be facilitated via customer profiling, up-selling, cross-selling, and the like. Customer's buying trends might be tracked, so that opportunities to sell further items can be capitalized upon. Up-selling, for instance, would include the marketing of a peripheral item to a buyer of a main component, (i.e. up-selling a scanner to a buyer of a computer). Cross-selling, for instance, would include targeting other types of selling (for different products, and the like) to the customer. The present invention readily provides this capability, as all such transactions flow through one system. This implicit one-to-one marketing cannot be achieved by other systems, which might require a user to switch between different servers to complete different transactions.

Context Sensitivity. The present system provides for information to be pulled up from the different information sources in a context sensitive manner. For example, if a user is researching routers, and security issues related thereto, the system will pull up other information relating to this topic, as well as products which need to be sold, in a context sensitive manner. The user will therefore not have to move from one site to another site in

order to research various aspects of router security -- instead the community capability of the present system provide all the information on one convenient page (or pages).

Inventory and Customer Service Assistance. The community aspect might also be used to assist a variety of participants in providing customer service assistance. The 5 community aspect might also assist in keeping up-to-date on inventories. For example, tools can analyze various schedules and problems associated with such schedules. If a car owner wanted to bring a car in for service, he might log onto the system and schedule an appointment. If any problems existed with the car, he might describe the same. A rule based analysis of these problems might generate a list of possible parts that might be 10 needed in order to fix the car. Other routines might go check the dealer's inventory for such parts. If any of the parts did not exist in that dealer's inventory, then this problem would be flagged for attention of the dealer. Apart from this particular dealer acting to solve the problem, an inquiry might be made to other dealers, and the parts automatically ordered, so that the list of potential parts would be on-hand when the customer came in for 15 their repairs. This would minimize any wait on the customer's part, and thereby build trust and loyalty of the customer for this dealer. The dealer is thereby supplied with added value by using the present system, because of the increased efficiency, and customer satisfaction.

Referring now to Figure 4B, a screen shot is shown of the overall Asera Portal, and 20 various example features offered.

Figure 4C shows a block diagram having various example vertical applications 1600. Certain community functions 1602 are illustrated, including message boards/chat functions 1604, industry news 1606, auctions 1608, ratings 1610, technical forums 1612, career development 1614, and complimentary services 1616.

25 Figures 4D-1 and 4D-2 show further features offered via example community screen(s).

PROFILE MANAGER. The profile manager is provided to track user behavior and recommend strategies for improvement of sales efficiency (e.g. cross-sell and upsell).

CONTENT AGGREGATOR. The content aggregator (or manager) 440 30 aggregates content from the many different information sources associated with the overall system. For instance, the system will produce (or possess, or acquire) certain knowledge. Such knowledge might be transaction related information, such as the level of sales

completed in any given month. Additionally, such knowledge might include (but is not limited to) any of the following: marketing information stored in an appropriate database; product information, data sheets. The knowledge might also include user information, such as frequently asked questions and their corresponding answers. Hence, this example
5 knowledge base might be generalized to include operational knowledge, transactional knowledge, marketing knowledge, and customer support information. The various content is stored in an appropriate location (or database) for such information. With the present invention, such content can be aggregated from diversified sources and presented to the user. For instance, a user might be interested in seeing a diverse set of information
10 including: a product description with pictures, comments regarding what people are saying about the product, frequently asked consumer questions, and a compilation of product related problems. This diverse set of information can be collected from the varied data stores associated with the system and presented to the user on a single page (or series of pages). To satisfy these functions, the content aggregator (or manager) includes two parts
15 -- content delivery and content display.

REPOSITORY. A repository 450 is used to store a variety of information, including the various metadata which is necessary to support the present architecture. The repository might consist of databases, data stores, and the like, for storing information such as rules, data, and objects. A lock manager 422 is further included in the core
20 services to manage the flow of data -- in a consistent and secure manner -- to and from the repository sources (or devices). Any time multiple people are using the data in the repository, it then becomes important to synchronize such activities.

For instance, the business objects associated with the formation of the business applications have rules, stored as commerce business rules 451. The template manager
25 uses rules, stored as template domain rules 461. Such rules might pertain to where an enterprise logo is placed, as well as storage of the logo itself. Security data 459 relates to identification and/or password information for each user that might log onto the present system. Profile and configuration data 455 relates, in part, to the community feature 408. Also, certain products might be offered on different machines, and the data must be
30 configured between these machines.

For any transaction that goes through the system, a set of log data 457 is stored. This transaction data might later be used by a business intelligence application (419) – through the profile manager – in order to perform data mining and the like. For instance,

the data might be mined in order to discover certain selling trends, buying trends, etc. Such data can later prove to be very valuable, and the present invention offers development and tracking of such data across many different (customized) applications.

The business objects (and steps) are stored as element 453. Such steps include 5 simple, interactive, and composite steps. The business objects (and steps) 453 are closely associated with the commerce business rules 451. In certain instances these two elements might be configured (or illustrated) as one single storage element, as the objects, steps, and rules are closely related.

DEVELOPMENT OF BUSINESS PROCESSES. Referring now to Figure 8, a 10 block diagram 800 is shown of certain representative elements relating to the development of a business process (or application). This development will generally involve three elements: business objects, business steps, and business rules. A business object can be defined as a logical representation of business data. Examples of such business objects include a sales order, a purchase order, a customer, and the like.

When an application 802 is written, it will access the data as a business object 804. 15 Line 806 represents a separation between the logical representation of data (on the left) and the physical representation of data (on the right). The business object 804 will undergo a mapping from the logical representation to the physical representation of such data, or a physical data source. A data source might be any device or resource that 20 generates data. In the present architecture, this might be a database table, an ERP system, or some other third-party software. An example in the present system might include the product configuration device, and/or any system or software that can generate data. The present system maps that data into the business object representation, and the application will manipulate the data through these business objects.

A series of adaptors (or plug-in adaptors), shown as 808, 810, and 812, are 25 thereafter used to map the data into the data source (e.g., an ERP source) 820. The business object can pull data from multiple adaptors. The adaptor will be responsible for converting any such data into a representation, or schema, for use by the data source.

The business object might also be mapped partially to several different adaptors. 30 For example, the business object can be comprised of an upper set of 5 fields 814, and a lower set of 5 fields 816. The first 5 fields can map to adaptor 808, and the bottom 5 fields can map to another adaptor 812. Additionally, the business object data can be

mapped to a second (underlying) adaptor 810 for communication with a second data source 822. Each adaptor can be designed to communicate with a different data source.

On the next level, a business process can be described as a workflow (or process flow). Referring now to Figure 9, a series of example steps in the form of a block diagram 900 are shown. The first step involves adding a line item 902. Step 904 is a price calculation, and step 906 inquires whether the product is available. If "yes", then step 908 shows the allocation of inventory. While each step has input and output ports, step 908 illustrates an example input port 910 and the output port 912. Step 914 shows a simple step of calculating the tax after the allocation of inventory. Step 916 is meant to indicate that any of a variety of other steps might also exist in this particular flow. If the inquiry in step 906 produces a response of "no", then a cross-sell/upsell step 918 is shown.

Step 918 is shown as a composite step 920, wherein a series of other steps comprise the overall composite result. Each step might be a hierarchical composite of other steps, hence referred to as a composite step. A functional box that performs only a simple calculation is hence referred to as a simple step. Each of the steps can have multiple input and output ports, and each of the ports can have multiple parameters. The parameters can also be different for each of the respective input and output ports. Accordingly, each functional box (or step) can have multiple functions, with each function operating on a different set of parameters. Alternatively, a box might perform only one function, but perform it differently depending upon the value of the parameters. Even a simple step will have multiple ports and parameters. The input/output ports and parameters for a composite step are determined as part of the interface design for that composite step.

A primary benefit of this approach is that a set of visual tools (for constructing and arranging the steps, ports, and parameters) can be readily imposed on the workflows. Additionally, this approach allows for ease of communication between an application developer and the consumer service personnel (or people). If the consumer service people want to add new functionality, for example, they can observe the overall flow of data from the composite step in relation to the entire flow. Functional boxes can be added, and the workflow can be quickly analyzed for data correctness.

Additionally, the present system provides Interactive Steps. An Interactive Step 911 is analogous to an input/output workflow application to the external world. It might generally involve a prompt to the user, i.e. a query about whether the user wants to buy a

certain product. Such a prompt might be done via a browser 912 or the like that interfaces with the user. An interactive step is replaceable. For instance, one configuration might support talking to a browser. However, a later configuration might replace the interactive step with an EDI (or similar) enabled capability. This latter configuration would thereby
5 be able to start communicating with the workflow through an EDI interface.

This example workflow can provide two different representations: 1) how the application is to be implemented; and 2) how the display navigation is being done from one screen to another. Any such implementation can involve an underlying separation of these two approaches. However, overall workflow is presented here as a generalized
10 result.

The business rules are next applied to the workflow or process by consumer (or customer) service personnel. Each of the rules has a name, and this name is generally used to identify particular functions that the rule will perform. A rule might serve as a callout to a particular application. A rule might also serve as an in-memory trigger on the business object. In the particular example of Figure 9, the product available inquiry (906)
15 might warrant a callout 930 for the warehouse 932 to be checked for such inventory. The consumer service personnel would use an appropriate rule, as applied to the existing workflow, in order to produce this result.

The use of the rule as a memory trigger involves affecting (or moving) the
20 associated trigger points. The business object itself represents a unit of data. At all times, when the application triggers the business object, the developer (or user) might want to perform data validation, or conversion, or perform some side effect on the business object. The process thereby identifies certain well known points where the business object is manipulated. For instance, the professional service person might desire to set a common
25 attribute of the business object. Alternatively, the professional service person can change the relationship between the business objects. These well defined points will serve as triggers.

Once the applications manipulate the business objects, there will be business rules
30 that can be associated with the trigger points. The business rule conditions will thereby serve as the triggers. For example, a trigger point might be defined as an attribute, with the triggering condition being the price limit of a purchase order for a particular user. Referring again to Figure 8, the application has set a first price amount 830. Through the application of a business rule, a validation might be performed on that trigger point.

Alternatively, a user might be limited to placing orders that are less than a certain amount, for instance \$1000. This might represent a second price amount 832. The trigger could be moved from the first point to the second point. The process (and applied business rule) would then verify the latter applied condition (rather than the former).

5 This latter description in Figures 8 and 9 corresponds to the above mentioned process of forming a business process (i.e. "stitching together" the various business steps) and then applying the various business rules to further customize the resulting application. While the discussion above speaks in terms of "moving" the trigger point, there is really only one physical point in memory. Two users might be limited by price in performing an
10 act (for instance). Both users arrive at the same point in memory. However, depending upon which user is contacting the memory point, a different result will ensue because of the different conditional limits associated with each user.

15 Rules can also be used for data conversion. For example, if a user logged in from a remote location (i.e. Japan), the data can be translated from yen to dollars. Such conversion might depend upon such conditional factors as the value, the user, and the user profile.

20 Generically rules can comprise any other customer specific function. For example, if an order is submitted, but the status of the purchase order is changed, an email message might be sent to a managing party regarding the status change. A rule can be created and applied to achieve this (or other) specific functionality. Generally, one rule specification exists per business object, and then graphical developer tools are used to implement the rule.

25 FORMATION AND USE OF TEMPLATES. Templates can be used to display certain information about the enterprise business, in a repetitive manner across many screens via a macro-template. Micro-templates can also be used to display information which might dynamically change. Still other wire frames might be used to dynamically link a function (or result) into a page.

30 Referring now to Figure 10, a block diagram 1000 of a web page is shown. Certain example labels 1002, 1004 are shown. These labels represent wireframes, which are slots for a link (or invocation) of other code. An example HTML label might include \$BCMS_FOO_BCMS\$, which might invoke a routine for area "foo" associated with the ACS. This label would be embedded into the HTML forming a page. A set of menus

1006 is shown to the left of the page -- this is a standard window that the user operably expects to encounter. A master-template 1008 is included, which would have enterprise business information such as a title, and/or logos. Together, the wire frames plus the master-templates form macro-templates. For each of the micro-templates (see below),
5 within the wire frame the back-end server side functions are used to generate HTML, on the fly, for each displayed screen.

Referring now to Figure 11, a catalog display step is shown, wherein this step is a composite step 1101. In this instance, the user desires to perform a search of catalogued steps (or objects), with a list of results being shown thereafter (as per element 1102). A
10 keyword 1104 is entered, which queries a database 1106. A set of results 1108 is displayed 1110 to the user/requester. The user then selects 1112 a particular entry, in this instance an "order entry" 1114.

Referring now to Figure 12, further features 1200 of the present invention are shown in relation to an example composite step. A composite step 1202 is further
15 comprised of simple steps 1203. As part of any general composite step 1202, an inquiry for information 1204 is sent to a database 1206. The results 1208 are thereafter sent back to the requester, and formatted using HTML, or the like. The formatted data is then sent to the client/user, who is likely viewing the data via a browser 1210. Now the persons using the browser can click-on (or select) some feature presented on the display, and then
20 a request 1212 is sent back to the server. At this point, the server might desire to perform validation of the data. Also, unmarshalling of name value pairs might be performed. For instance, certain data might come back with some end quota, but such data needs to be made useful via unmarshalling of the data. Pure JAVA coding (in this example) might be considered a simple step.

25 A micro-template is, by definition, a place holder for an HTML snippet within an HTML page to dynamically embed and generate data coming from any data source, irrespective of protocols, and can be rendered in a language independent manner. Hence a micro-template might be considered as "real estate" on the HTML page. The micro-templates can be made drag-and-droppable using a standard editor. The micro-template is
30 declared in a macro-template as a group (a wireframe), and the macro-template can be instantiated using an master-template. To have a set of logos (or the like) appear consistently across every displayed page, a master-template will be supplied by the enterprise business that desires to present a consistent look-and-feel across many pages.

If (for instance) a website had many pages, the master-template would be propagated across those many pages. Each macro-template can have a group of micro-templates that will be used to compile information from different sources during run-time.

The interactive step 1214 is broken down into "pre" "main" and "post" processing segments. The "pre" segment involves certain JAVA code, which tells the step how to pre-populate certain data areas. The "main" area is where data is actually shipped and persons can declare what goes where, and the "post" segment contains JAVA code to process user input.

One advantage of the present invention is that some of the static parts of a template are generated from pre deployment environment, wherein a dynamic part might come from any type of independent dynamic source. The templates can also be rendered using any type of language. For example, present applications use the JSP/ASP model. In JSP, a programmer can use only JAVA. In ASP, a programmer can use only virtual basic, or JAVA script. In the present case, creativity is separated from programmability.

Generally, a web page is created by an artist. The programming however is done by a technical (IT) person. If ASP is used, the same person must be skilled in both respects. Under the present system, the creative person can concentrate on creating the macro-template, wherein the technical person can concentrate on creating the micro-templates. During runtime, both of the results will be merged. The various templates might be rendered in JAVA, or XSL, or straight URLs. Implementations might also be provided for use of "C" or visual basic or other such languages.

The wire frames are intended to include a group of micro-templates to make a meaningful application. Under the present invention, these wireframes turn out to be a foundational element of each application. These wire frames can be taken and dropped into any other webpage to achieve the same application and/or functionality, with minimal effort. In large part, this is because the applications are language independent. For instance, one website can be customized to another party's website with few additions, and regardless of the language used for either website.

TOOLS FOR FORMING APPLICATIONS. To create ACS applications, application engineers must create several types of artifacts: 1) business objects; 2) business object configurations; 3) business steps, including a) simple, b) interactive, c) request, and d) composite; 4) Wire frame files; and 5) Master-template files. A request step is a callout to another step, and is analogous to a "goto" programming step. The first

three types of artifacts are encoded in XML files, and the last two are in HTML. In addition to being XML files, such objects, configurations, and steps can also refer to each other, hence an XML file may not be independent.

An application engineer (or professional services engineer, or consumer service person) can create and edit XML files using a text or XML editor. However, this process can often be tedious, error-prone, and time consuming. In addition, certain types of artifacts such as composite business steps (which are used to create workflows) are most naturally described in a graphical (diagrammatic) manner. Furthermore, even relatively simple artifacts such as business objects can be relatively complex to describe in XML, but can be more easily described using a customized graphical editor.

An ACS application designer/builder (henceforth referred to as the tool) is an integrated development environment (IDE) for creating and manipulating ACS artifacts, and consists of a general extensible framework and a set of custom plug-in "editors" -- one for each type of artifact. The tool performs additional input validation when loading XML artifacts, thereby identifying errors very early in the design process. Furthermore, each custom editor, to a large extent, also prevents the user from creating ill-defined or invalid artifacts.

The tool architecture is configured so that new object types can easily be added through the menu structure. Object repositories can be both local (file system) and remote (HTTP). The tool can run stand alone in a computer, or in a web browser. The plug-in editor framework also uses an extensible tabbed view metaphor. For example, new objects might easily be added, with each new object carrying a set of edit tabs. The composite workflow editor displays complex business application process flows in a simple manner. This allows users to compose new workflow logic from existing components. The business object configuration includes a set of editors that allow a user to quickly customize business objects for specific customer deployments. Also included are default filter editors, custom filter editors, and named rules. Integrated tools include the ability to compile and build a feature on the fly. A wire-frame can also be merged onto the resulting application so that it can be invoked from a display screen (see template discussion).

Referring now to Figure 13, a series of application designer screens are shown that further demonstrate the functional implementation of the above mentioned features. In this example, the materials are grouped on a project basis. Figure 1300 shows a window

1302 with a project tree window 1304, a message window 1306, and a tabbed editor window 1308. The project tree window 1304 lists all of the items in the project. Unique icons and a folder hierarchy are used to identify and group different types of artifacts. These artifacts might include business objects, business object configurations, business 5 steps, wireframes, and master-templates. The message window 1306 lists the various warning messages, error messages, and status messages, etc. These warning messages assist the developer in stitching together the various steps to form an application. The various parameters associated with the input and output ports might be constrained by types (i.e. real or integer values). In order to link the various boxes (or steps), the problem 10 becomes more than simply linking one box to another with a connector. The flow of data across the various parameters associated with each step must also coincide. The ports must also be validated as compatible (i.e. not every port can be connected to some other port). For instance, one port might expect three integers, whereas another port might expect three string variables. The graphical tool allows new ports to be added or deleted, 15 and similarly new parameters to be defined for any given step.

If certain input ports, output ports, and related data parameters in each of these ports do not properly coincide, then error messages will help facilitate any corrections to the workflow. Window 1308 shows a tabbed editor which allows for graphical implementation and placement of the various steps. The steps are stored and listed 20 according to a catalog of such steps, wherein the catalog can be searched. A collection of pre-defined steps exists for performing certain standardized tasks. For instance, displaying certain information, or tabulating certain data involves the same basic logic.

The processor type (i.e. JAVA, XSL, URL, etc.) can also be specified, by developer choice, for each micro-template of an interactive step. This allows for added 25 flexibility when dealing with different development languages, across many different systems.

Hence, generalized routines (or steps) can be called upon by other routines (or steps). Ultimately, a catalog, or palette, of pre-defined and tested steps can be used by a developer to build business routines. This makes forming new routines much easier and 30 accessible, especially for non-technical professionals.

Each item in the tabbed editor window can be edited via a number of item specific tabs. In this example, a composite step can be edited via three tabs 1310, 1312, and 1314. The selected tab (workflow 1314) displays the workflow editor used to assemble the step

instances, and transitions, of the selected composite (or simple) step. In this particular embodiment, the steps are indicated as simple by the preceding "s" in their names, and composite steps use a preceding "c" in their names. The various connectors 1316 show the data flows between steps. Graphical icons 1318 are used to indicate certain aspects of the parameters being passed within the connection. Tab 1310 can be used to show the various input ports for a particular step (composite or otherwise). Tab 1312 can be used to show the various output ports. By quickly pulling up the various data interface requirements between the steps, a working application (i.e. error free, with no data conflicts) can be constructed.

10 Display filters might also be used to quickly change various aspects of the display. For instance, the text being displayed might be associated with certain fields in the filter. The developer can therefore change only this field and not affect the overall functionality of the developed routine.

15 The aforementioned rules can also be applied via the graphical interface. In one sense, the rules act as database triggers. The rules generally provide conditional directions regarding what to do with certain objects (see description above).

20 Referring now to Figures 13A-130, a set of example screens is shown relating to the GUI tool. In Figure 13A, a screen is shown where an example "product" object is selected in the project tree window 1304. A set of tabs 1322 (i.e. implementation, attributes, relations, and access operations) are shown for this object selection. Other types of objects might utilize a different set of tabs. The implementation tab 1324 is shown selected. This results in main window 1325 showing JAVA code for a method which is defined in association with that object.

25 Figure 13B next shows the "attributes" tab 1326 being selected. The attributes are listed by name and the properties are shown. An attribute typically has domain, a cardinality, and key. The attribute might be flagged as required or not. The attribute might also be persistent or not. A comment section 1327 is shown which might show descriptive comments to further document why/how each field is used.

30 Figure 13C shows the "relations" tab 1328 being selected. This screen allows the present invention to show relations between various objects. For example, an employee object would have a relation to a manager object, and so forth.

A business object configuration is generally used to customize business object for each customer. In this example, the business object configuration has three parts: default filter, custom filters, and named rules. Figure 13D shows an example business object configuration for "Product" 1332. This configuration is stored in a configuration catalog 1330 as viewed on the project tree list. The default filter selection 1335 shows a set of attributes/relations 1334 which are in use for each selected object configuration. A set of unused attributes/relations 1336 are also shown. The attributes can be moved from used to unused states via the add/remove filter buttons. The attributes of the selected "Price1" 1338 are shown as a named pair including format 1340 and currency 1342. As another example, Dates (not shown as a selection) might carry long or short date formats, etc. The edit rules button 1344 allows a user to create and modify implicit rules associated with an attribute/relation. (as similar to Figure 13H, below).

Figure 13E shows the "custom filters" tab 1346 being selected. A list of custom filters is shown in the main window, with the "Productfeatures" filter selected. The buttons 1349 allow filters to be added, removed, edited, and renamed. Figure 13F shows selection of the edit filter button 1350. Window 1352 shows the attributes/relations in this particular filter. The up/down buttons can be used to toggle through the listing. Window 1354 shows the attributes/relations not in the filter. Window 1358 shows properties/values associated with the selected item. Special (non-attribute) items (e.g. 1360) can also be listed and used to control how an object will be displayed (i.e. color attributes, and the like). Items 1360 can generally be used to control a subset of attributes.

Figure 13G shows selection of the "named rules" tab 1362. A quote configuration 1364 is selected in the project tree window. A list of named rules is shown in the main window associated with the quote configuration. A "CalculateLinesDiscount" rule 1366 has been selected. By selecting the edit button 1368, the resulting Figure 13H shows the example code 1370 (XML or otherwise) which would complete the task defined by this rule. In this case, the rule performs a simple calculation. Many other such rules (and rule complexities) might exist, which would be tied to different events.

Example screens for manipulating steps are next shown. Steps involve inputting certain data and producing a result, or output data. The tabs for manipulating steps include implementation, input ports, and output ports. Figure 13I shows a listing of steps in the project tree listing window 1372. In this example, a simple step "sGetBOMRelations" 1374 has been selected. The implementation tab 1376 has been selected. This results in

JAVA code being displayed in the main window 1378. Figure 13J shows selection of the input port tab 1380. While many ports might exist, this example shows one input port "GetBOMListPort" 1382. Figure 13K shows the result of selecting "edit" 1384. The parameters for this input port are shown, with the name and type (i.e. integer, string, etc.) of each parameter.

An interactive step provides pre, main, and post implementation. As previously mentioned, interactive steps display output for, and obtain input from, the user. For example, the interactive step might show a static display and/or elicit data input from the user. In this example, the "iShowProdDetails" interactive step 1386 has been selected.

10 The pre implementation ("Pre Impl") tab 1388 results in the display of JAVA code associated with this pre implementation step.

Figure 13M shows the result of the main implementation step being selected. A macro template name 1392 is listed, along with a list of associated micro-templates 1394. The micro-template "MTProductFeatures" 1396 has been selected, and shows the 15 associated type and object name fields. This micro-template is associated with a processor name 1395 and processor value 1396.

Figure 13N shows the post implementation tab 1398 selected. JAVA code is displayed in the main window for handling the result of the main interactive step. For example, this code would handle the placement of data on an output port in response to the 20 prior user input (if user input has been so provided).

Figure 13O shows a workflow tab 1399 being selected for the composite step "eProductCatalogMain" 1397. The graphical workflow display shows various composite and interactive steps being connected. Example connector 1395 indicates a single transition between steps. Connector 1393 indicates a multiple transition in both directions. 25 Connector 1391 indicates a single transition in either direction. Color coding is used in association with the connectors to indicate activity. This further aids in development of a workflow.

Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims. Therefore, the described embodiments 30 should be taken as illustrative and not restrictive, and the invention should not be limited

to the details given herein but should be defined by the following claims and their full scope of equivalents.

CLAIMS

What is claimed is:

1. A method for providing custom configurable service applications from a standardized set of components associated with a server device, the method comprising:
 - 5 configuring best-of-breed components and associated server device applications for communicating with systems external to the server device;
 - forming and utilizing a first set of objects which are data representations for interacting with the systems external to the server device;
 - constructing a service application from a modular set of application steps and
 - 10 application rules to perform services according to user requirements, the service application interfacing with the best-of-breed components; and
 - forming and utilizing at least a second set of objects which are data representations for interacting with a user through the service application.
2. The method of Claim 1, further comprising:
 - 15 storing and retrieving data and information from a data repository, whereby the data repository facilitates the centralized storage, retrieval, and sharing of such data and information between applications.
 3. The method of Claim 1, wherein the custom configurable service applications includes at least business service applications, and the objects are business
 - 20 objects.
 4. The method of Claim 3, wherein at least one business service application includes a product catalog for an enterprise business.
 5. The method of Claim 4, wherein at least one business service application also includes a product configuration manager for configuring a product from the
 - 25 enterprise business.
 6. The method of Claim 3, wherein the set of application steps are business steps that include operations with a defined set of input and output ports, with each port having a defined set of parameters.
 7. The method of Claim 6, wherein the application rules are business rules,
 - 30 which are used to capture customer specific business practices.

8. The apparatus according to Claim 7, wherein an existing business service application can be selected, further developed, and customized.

9. The method of Claim 7, wherein the business rules are used to produce different results depending on conditional triggers that are associated with the underlying
5 business objects.

10. The method of Claim 7, wherein the constructing step further utilizes a graphical user interface for arranging the business steps and business rules comprising a business service application, with data connections therebetween.

11. The method of Claim 10, whereby feedback is provided as to the validity of
10 the data connections between the arranged business steps and business rules comprising a business service application.

12. The method of Claim 1, wherein the best-of-breed components and underlying applications comprise and run the commerce server.

13. The method of Claim 7, wherein the best-of-breed components are
15 interfaced with the business service applications through APIs.

14. The method of Claim 13, further comprising: providing a catalog of APIs for developers to reference in developing applications that interface with the best-of-breed components.

15. The method of Claim 7, further providing adapters for communicating with
20 the back-end external systems.

16. The method of Claim 15, wherein the back-end external systems include ERP systems.

17. The method of Claim 15, wherein a separate adapter is provided for each different external system.

25 18. The method of Claim 1, further providing a community enabler for accessing data and information from different functional components of the server device, external systems, and service applications, whereby the user can request an aggregate presentation of such different data or information into at least one display area.

30 19. The method of Claim 18, wherein the user can transition back and forth between the different data or information without having to connect to another server device or website.

20. The method of Claim 18, wherein the user is provided with a single source log-on.

21. The method of Claim 18, wherein templates are used to comprise a page, with the data or information dynamically linked into the page from a corresponding data source.

5 22. The method of Claim 21, wherein a master-template is applied across a series of pages, and a micro-template is used to selectively display data from a variety of sources.

10 23. The method of Claim 18, wherein the data or information is accessed in a manner which is context sensitive to at least part of the data or information.

24. The method of Claim 1, wherein the service applications are provided to a user through a network connection.

25. The method of Claim 24, wherein the network connection is an Internet connection.

15 26. The method of Claim 1, wherein the service applications are provided to a user through a disconnected mode.

27. A method for providing custom configurable business service applications to a user through a server device, the method comprising:

configuring the server device as comprised of best-of-breed components;

20 configuring the server device with best-of-breed server applications for interacting with the best-of-breed components and providing server functionalities;

constructing a business service application from a modular set of components that can be dynamically arranged according to user requirements, the business service application interacting with the server applications to provide the user with the server 25 functionalities.

interfacing with systems external to the server device via adapters configured to communicate with the external system.

28. The method of Claim 27, further providing for:

storing and retrieving data representations from the business service application and the server applications in a centralized repository, whereby the data representations are readily available to all functions and applications associated with the server device.

29. The method of Claim 28, further providing:
5 enabling the community exchange of data and information between all applications and server functions associated with the server device.

30. The method of Claim 29, wherein the user can request an aggregate presentation of the different data or information into at least one display area.

31. The method of Claim 29, wherein the modular set of components includes
10 business steps which are operations with a defined set of input and output ports, with each port having a defined set of parameters, and business rules which are used to capture customer specific business practices.

32. The method of Claim 31, wherein the business rules are used to produce
15 different results depending on conditional triggers that are associated with the underlying business objects.

33. The method of Claim 31, wherein the constructing step further utilizes a graphical user interface for arranging the business steps and business rules comprising a business service application with data connections therebetween.

34. The method of Claim 31, wherein the step of interfacing with systems
20 external to the server device utilizes at least XML.

35. The method of Claim 31, wherein data representations between the business service applications and users are formulated and managed as business objects.

36. The method of Claim 35, wherein data representations between the server device and systems external to the server device are formulated and managed as business
25 objects.

37. The method of Claim 36, wherein the business rules are used to produce different results depending on conditional triggers that are associated with the underlying business objects.

38. The method according to Claim 27, wherein an enterprise company can
30 quickly acquire a turn-key e-commerce solution to automate their channel relationships.

39. The method according to Claim 27, wherein a net market maker can quickly acquire a turn-key e-commerce solution to automate their channel relationships.

40. An apparatus for providing custom configurable service applications from a standardized set of components associated with a server device, the apparatus comprising:

5 a server device configured from best-of-breed components, and running best-of-breed server applications to provide certain server functionalities, the server device having server data including data representations for interacting with systems external to the server device, the data representations being managed as objects;

10 a graphical design tool used for constructing at least one service application from a modular set of application steps and application rules, the service application performing services according to user requirements, the service application being capable of interfacing with the best-of-breed components and server applications, the server data further including data representations for interacting with users, the data representations being managed as objects;

15 at least one adapter device for interfacing with the systems external to the server device; and

a repository for storing and retrieving server data, the repository facilitating centralized storage, retrieval, and sharing of server data between all the server device components, server applications, and service applications.

20 41. The apparatus according to Claim 40, wherein the custom configurable service applications includes at least business service applications, and the objects are business objects.

42. The apparatus according to Claim 41, wherein at least one business service application includes a product catalog for an enterprise business.

25 43. The apparatus according to Claim 42, wherein at least one business service application also includes a product configuration manager for configuring a product from the enterprise business.

30 44. The apparatus according to Claim 41, wherein the set of application steps are business steps that include operations with a defined set of input and output ports, with each port having a defined set of parameters.

45. The apparatus according to Claim 44, wherein the application rules are business rules, which are used to capture customer specific business practices.

46. The apparatus according to Claim 45, wherein an existing business service application can be selected, further developed, and customized.

5 47. The apparatus according to Claim 45, wherein the business rules are used to produce different results depending on conditional triggers that are associated with the underlying business objects.

10 48. The apparatus according to Claim 45, wherein the design tool further utilizes a graphical user interface for arranging the business steps and business rules that comprise a business service application, with data connections therebetween.

49. The apparatus according to Claim 48, whereby the graphical tool provides feedback as to the validity of the data connections between the arranged business steps and business rules comprising a business service application.

15 50. The apparatus according to Claim 49, whereby the graphical user interface tool accounts for the ports and parameters between connected steps.

51. The apparatus according to Claim 47, wherein the best-of-breed components are interfaced with the business service applications through APIs.

20 52. The apparatus according to Claim 51, which further provides a catalog of APIs for developers to reference in developing applications that interface with the best-of-breed components.

53. The apparatus according to Claim 41, wherein the external systems are back-end systems.

54. The apparatus according to Claim 53, wherein the back-end systems include ERP systems.

25 55. The apparatus according to Claim 40, wherein a separate adapter is provided for each different external system.

30 56. The apparatus according to Claim 40, further providing a community enabler for accessing data and information from the different functional components of the server device, the external systems, and the service applications, whereby the user can request an aggregate presentation of such different data or information into at least one display area.

57. The apparatus according to Claim 56, wherein the user can transition back and forth between the different data or information without having to connect to another server device or website.

58. The apparatus according to Claim 56, wherein the user is provided with a single source log-on.

59. The apparatus according to Claim 56, wherein templates are used to comprise a page, with the data or information dynamically linked into the page from a corresponding data source.

60. The apparatus according to Claim 59, wherein a master-template is applied across a series of pages, and a micro-template is used to selectively display data from a variety of sources.

61. The apparatus according to Claim 56, wherein the data or information is accessed in a manner which is context sensitive to at least part of the data or information.

62. The apparatus according to Claim 40, wherein the service applications are provided to a user through a network connection.

63. The apparatus according to Claim 62, wherein the network connection is an Internet connection.

64. The apparatus according to Claim 40, wherein the service applications are provided to a user though a disconnected mode.

65. A server system for providing custom configurable business service applications to a user through a server device, the apparatus comprising:

a server device as comprised of best-of-breed components;

the server device configured with best-of-breed server applications for interacting with the best-of-breed components and providing server functionalities;

25 a construction tool for developing a business service application from a modular set of components that can by dynamically arranged according to user requirements, the business service application interacting with the server applications to provide the user with the server functionalities; and

30 an interface between the server device and systems external to the server device which utilizes adapters configured to communicate with the external systems.

66. The server system according to Claim 65, further providing for a centralized repository, whereby data representations from the business service applications, the server applications, the external systems and the users are stored and retrieved in the centralized repository, whereby the data representations are readily available to all functions and applications associated with the server device.

5 67. The server system according to Claim 66, wherein the repository is external to the server device.

68. The server system according to Claim 66, wherein the repository is closely associated with the service device.

10 69. The server system according to Claim 68, wherein further provided is a community enabler that facilitates the community exchange of data and information between all applications and server functions associated with the server device.

70. The server system according to Claim 69, wherein the user can request an aggregate presentation of the different data or information into at least one display area.

15 71. The server system according to Claim 69, wherein the modular set of components includes business steps which are operations with a defined set of input and output ports, with each port having a defined set of parameters, and business rules which are used to capture customer specific business practices.

72. The server system according to Claim 71, wherein the business rules are used to produce different results depending on conditional triggers that are associated with the underlying business objects.

73. The server system according to Claim 71, wherein the design tool further provides for a graphical user interface for arranging the business steps and business rules comprising a business service application with data connections therebetween.

25 74. The server system according to Claim 71, wherein the interface with systems external to the server device utilizes at least XML.

75. The server system according to Claim 71, wherein data representations between the business service applications and users are formulated and treated as business objects.

76. The server system according to Claim 75, wherein data representations between the server device and systems external to the server device are formulated and treated as business objects.

77. The server system according to Claim 76, wherein the business rules are used to produce different results depending on conditional triggers that are associated with the underlying business objects.

78. The server system according to Claim 67, wherein an enterprise company can quickly acquire a turn-key e-commerce solution to automate their business channel relationships.

79. The server system according to Claim 67, wherein a net market maker can quickly acquire a turn-key e-commerce solution to automate their business channel relationships.

80. A method for providing customized business application solutions to at least one enterprise company that contacts a centralized provider of such solutions, the method comprising:

assembling a collection of best-of-breed components to provide a seamless connection to services that are outsourced from the centralized provider;

capturing the particularized business processes of the at least one enterprise company;

providing a modular approach towards building a business service application for the at least one enterprise company; and

customizing the modularly built business service application according to the captured business processes.

81. The method according to Claim 80, wherein the enterprise company thereby acquires a turn-key e-commerce solution to automate channel relationships.

82. The method according to Claim 80, wherein the step of capturing the business processes includes determining the business processes in an automated manner.

83. The method according to Claim 82, wherein the step of capturing the business processes further includes determining the business processes in a human interactive manner.

84. The method according to Claim 80, wherein the step of providing a 5 modular approach includes managing the interactions with external systems or users as business objects.

85. The method according to Claim 84, wherein the business objects include data representations in an XML and Java combined format.

86. The method according to Claim 84, wherein the step of providing a 10 modular approach further includes utilizing business steps and business rules to construct the business application.

87. The method according to Claim 86, whereby the business steps are business operations with a defined set of input and output ports, each port having a defined set of parameters.

88. The method according to Claim 87, whereby the business rules are used to 15 capture customer specific business practices.

89. The method according to Claim 86, wherein an additional step includes: providing tools that allows the business steps to be arranged according to the needs of the enterprise business.

90. The method according to Claim 89, wherein the tools include a graphical 20 user interface that allows a developer to visually arrange business steps into business processes or workflows.

91. The method according to Claim 90, wherein the tools further allow the developer to associate business rules with the arranged business steps.

92. The method according to Claim 80, wherein an additional step includes: 25 providing community aspects between developed applications and functions offered by the centralized provider.

93. A business method for providing a user with a centralized framework for developing and customizing a business service application from a modular set of components associated with a centralized device, the method comprising:

- configuring the centralized device with best-of-breed components for
- 5 communicating with sources external to the centralized device;
- capturing the particularized business processes of a user;
- providing certain modular components that can be arranged to form a business service application; and
- 10 customizing the arrangement of the modular components to form a business service application according to the particularized business processes of the user.

94. The business method of Claim 93, wherein the user includes an enterprise company that can interact with the centralized framework to thereby create e-commerce business solutions without having to develop, maintain and upgrade the solutions in-house.

95. The business method of Claim 93, wherein the modular components include business objects as data representations in XML and Java combined formats.

96. The business method of Claim 95, wherein the modular components further include business steps as business operations with a defined set of input and output ports, each port having a defined set of parameters.

97. The business method of Claim 96, wherein the modular components further include business rules as conditional triggers associated with underlying business objects.

98. The business method of Claim 97, wherein the modular components can be moved, or the connections modified, in order to create and customize a business process.

99. A business method for providing a plurality of standardized business service applications and customized business service applications through a central server device, the method including:

- configuring the server device from best-of-breed components;

- configuring the best-of-breed components to communicate with data sources external to the server device via application interfaces (APIs);
- determining the business processes of a user business that contacts the central service device;
- 5 forming a customized business service application from existing standardized business service applications as modified by modular components provided by the central server device; and
- providing for community interaction between business service applications, and the central server device.
- 10 100. The business method of Claim 99, wherein the modular components include:
- business objects as data representations;
- business steps as business operations with a defined set of input and output ports, each port having a defined set of parameters; and
- 15 business rules as conditional triggers associated with underlying business objects.

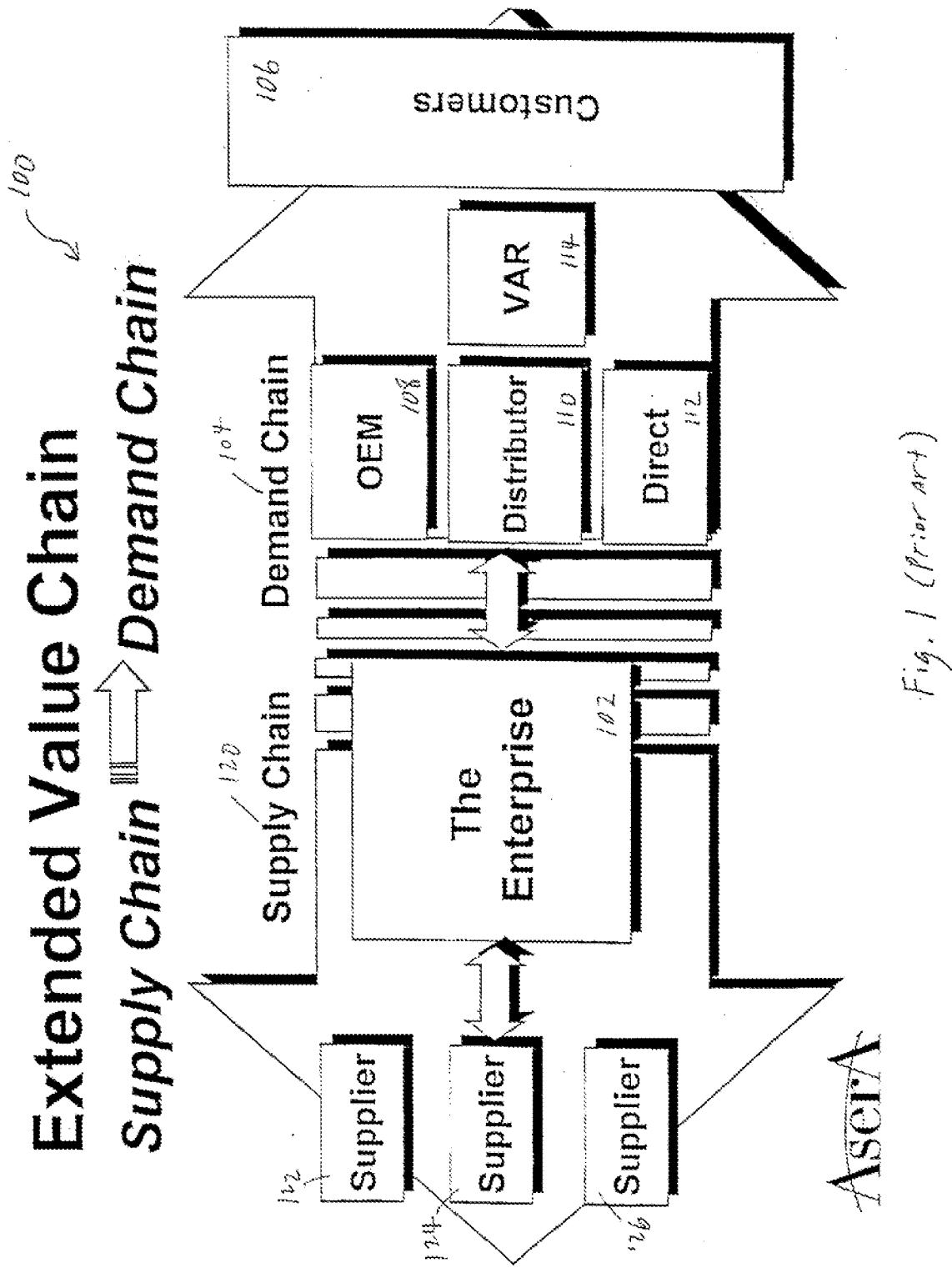
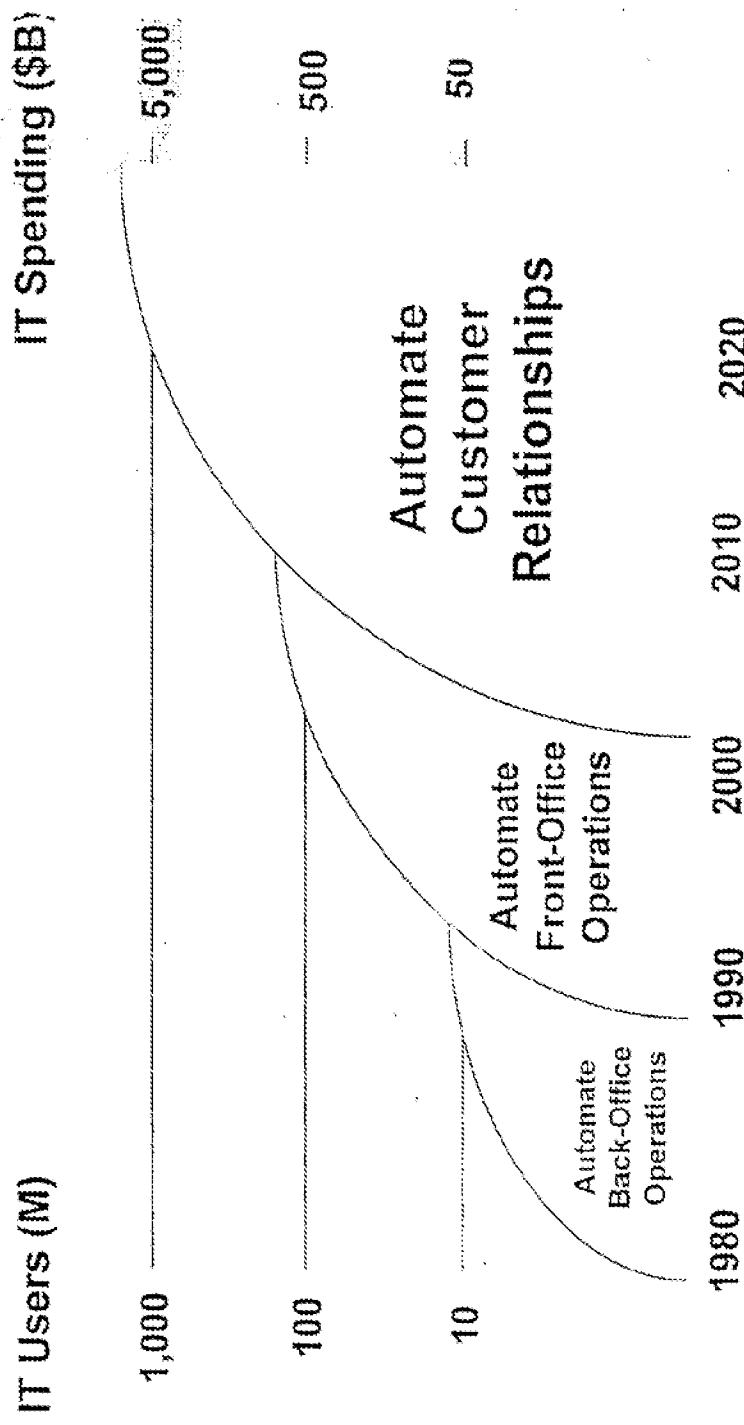


Fig. 1 (prior art)

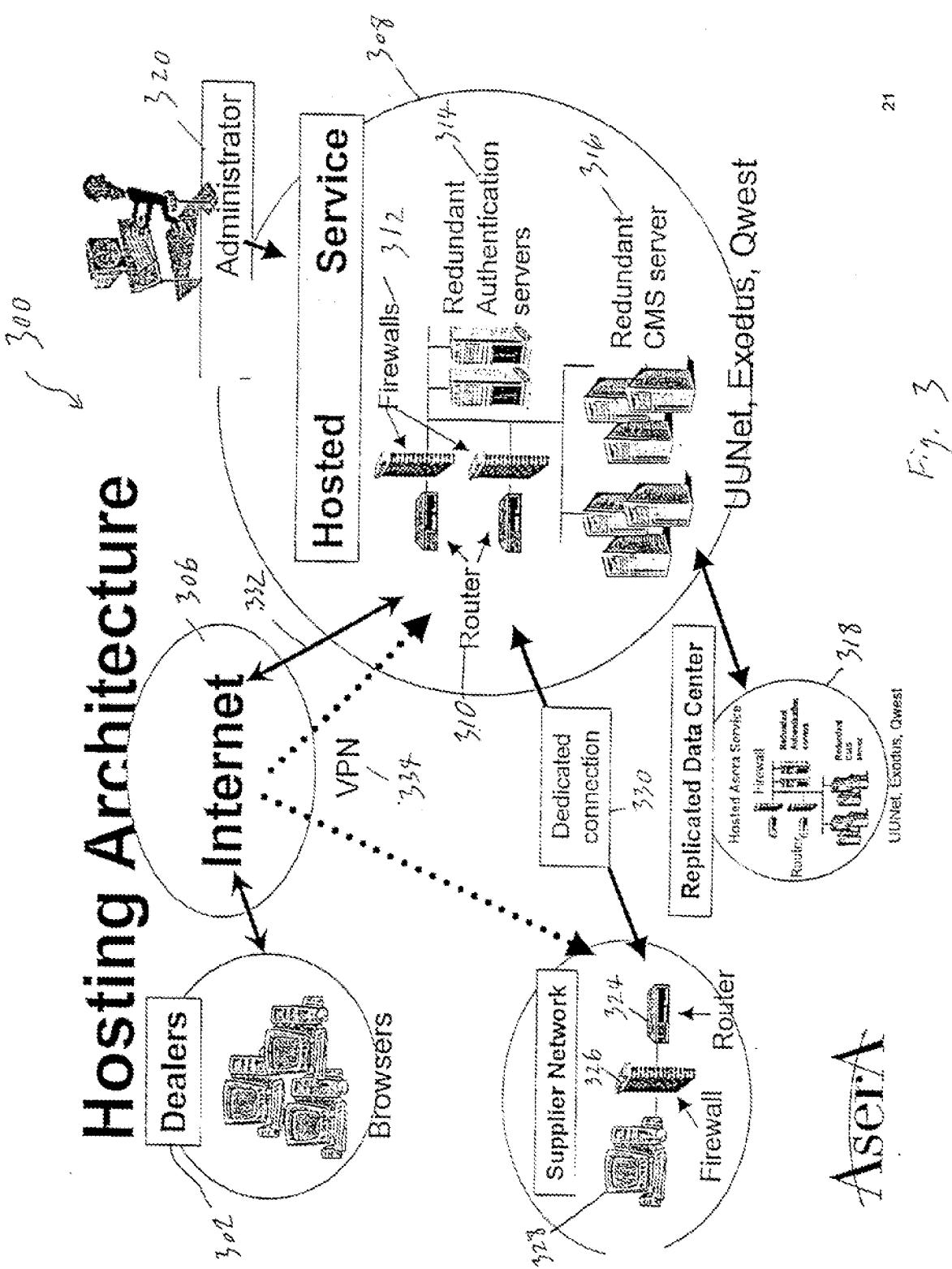
The Enterprise IT Agenda

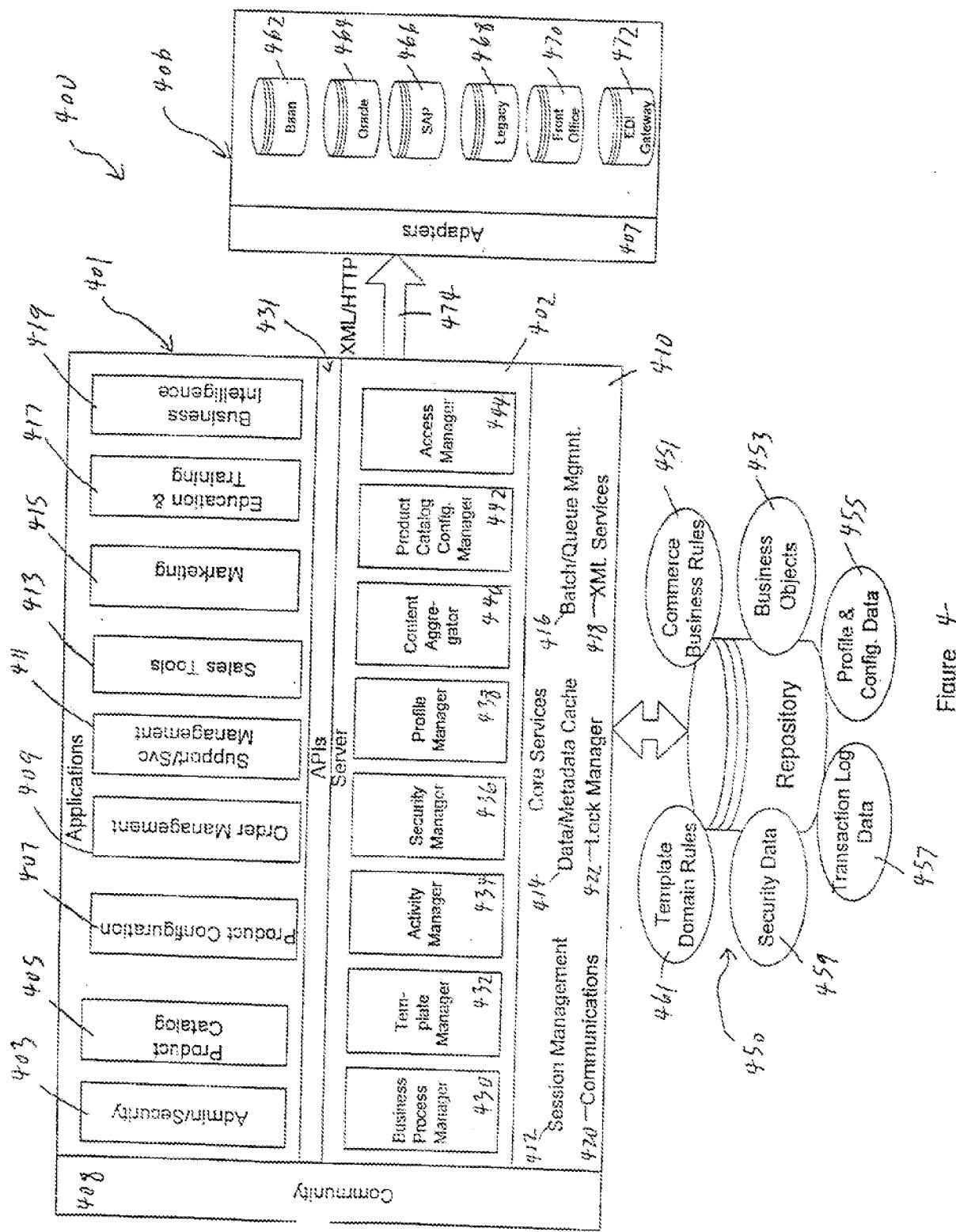


Source: Frank Gens, Chief Analyst, IDC

Fig. 2 (Prior art)

ASeia





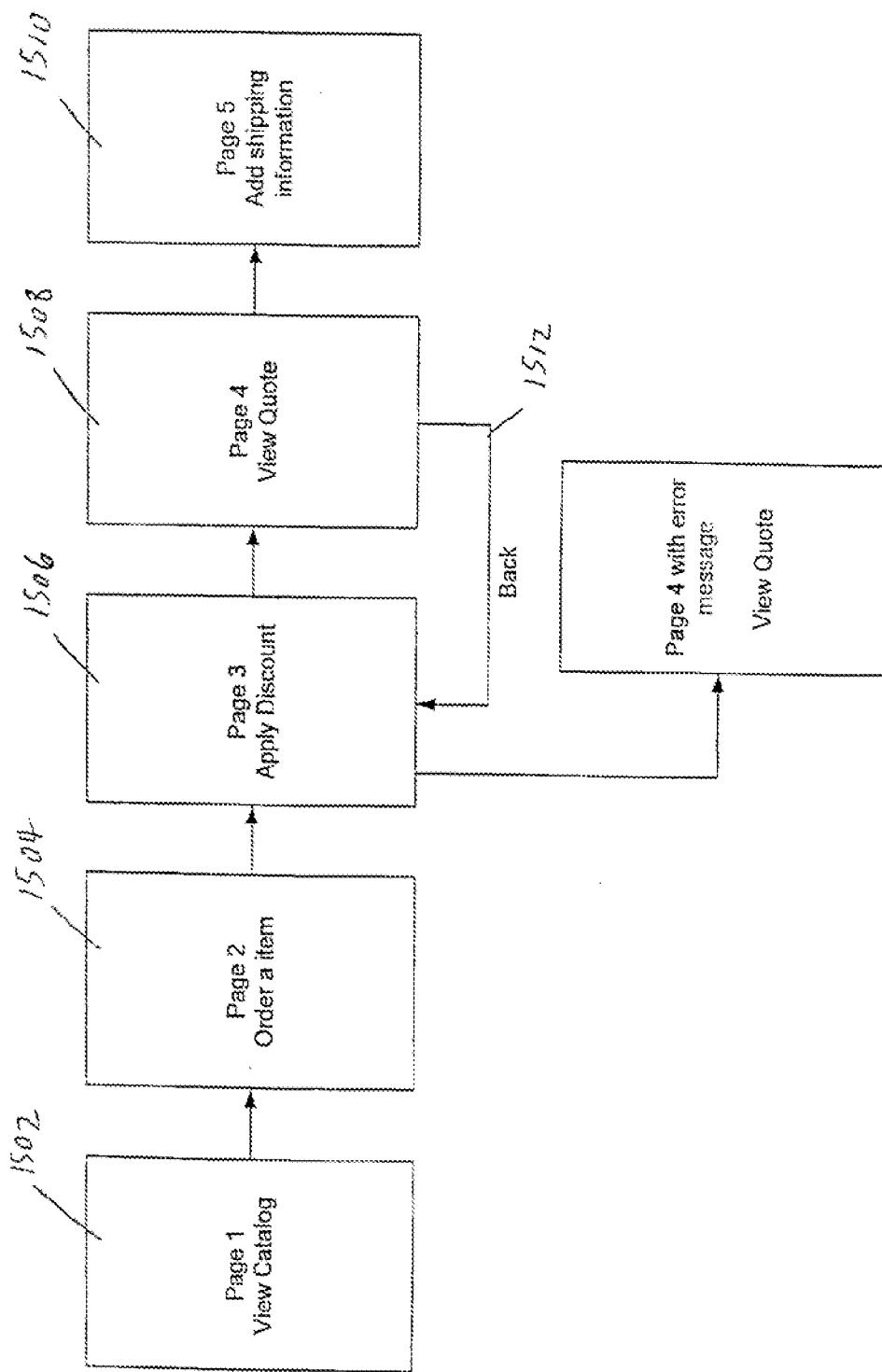
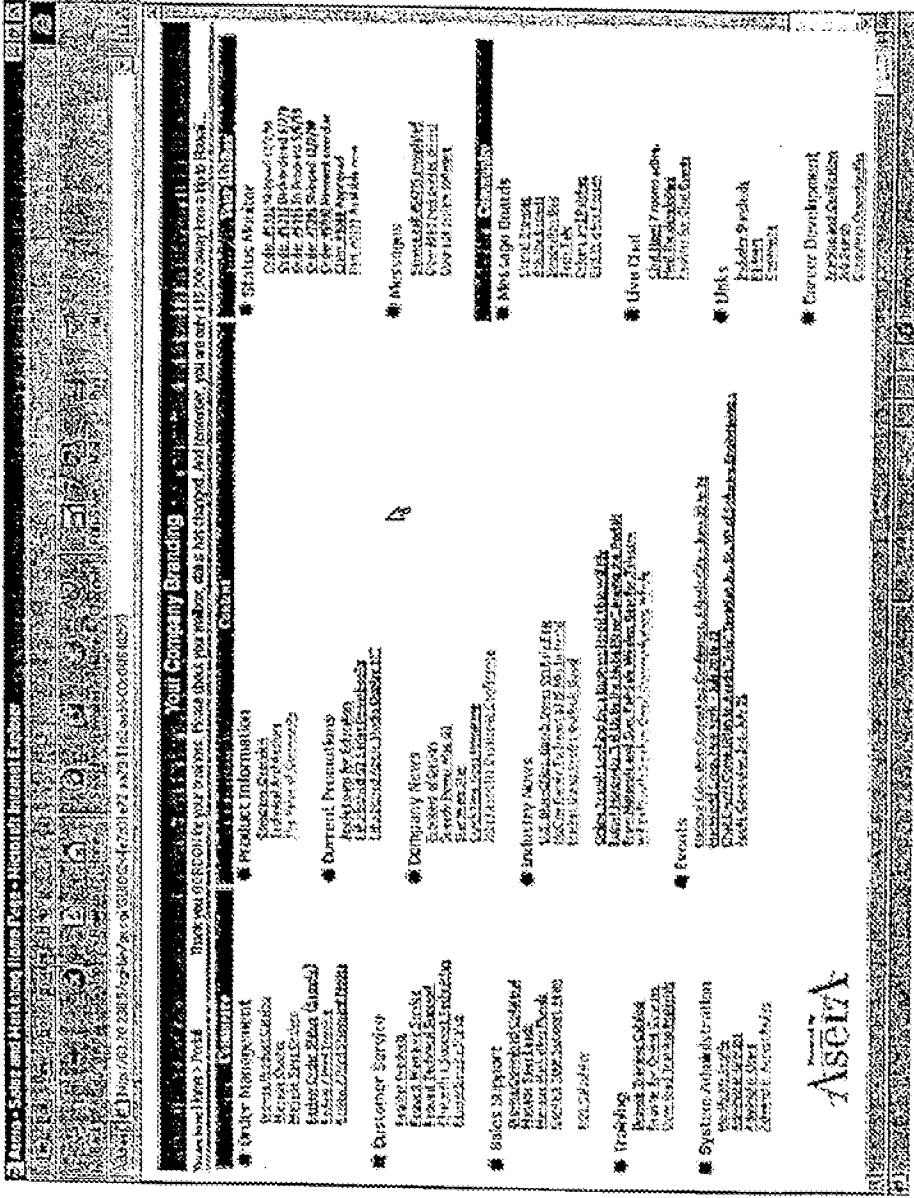


Figure 4-9

The Asera Portal



6/37

Channel Management Service

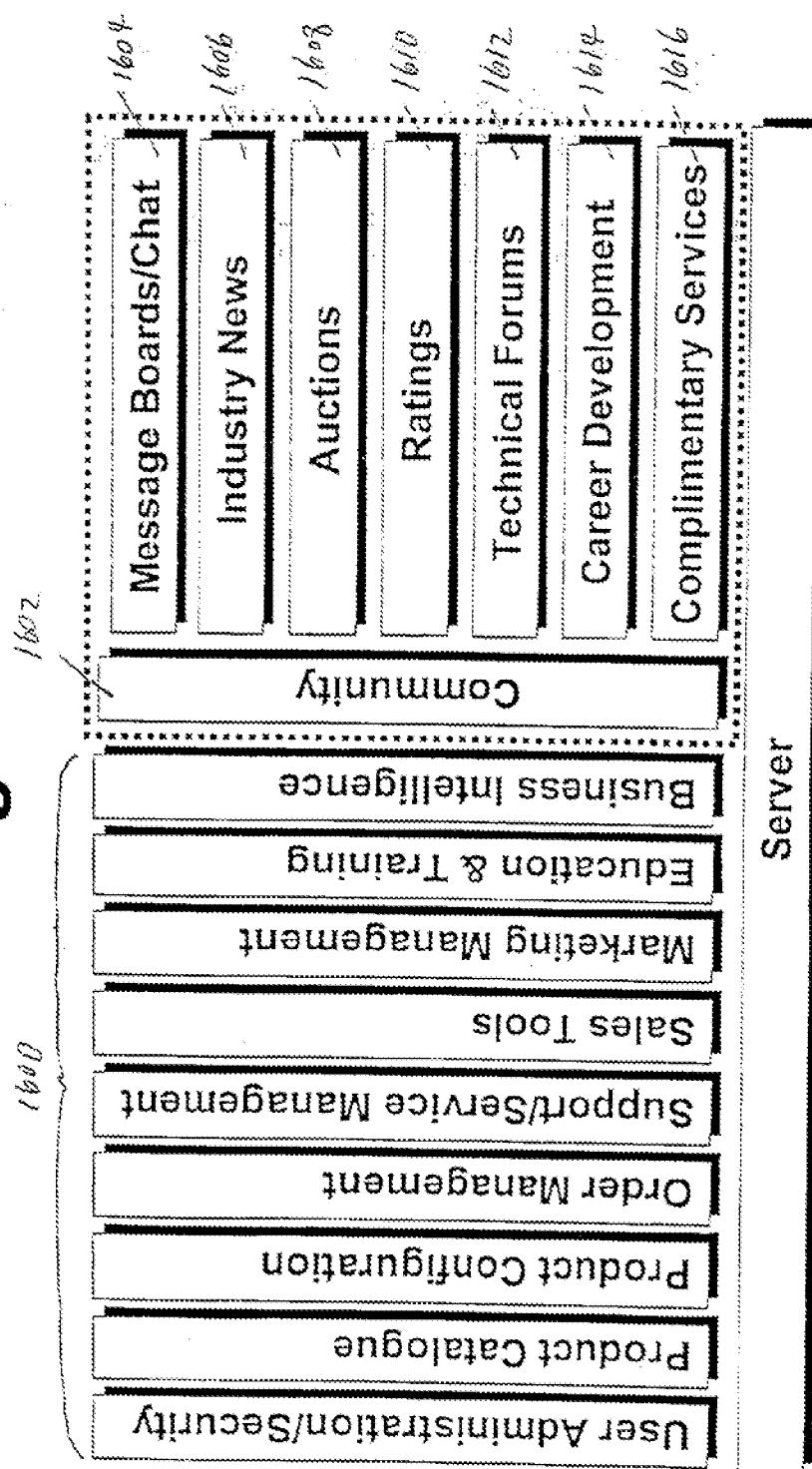


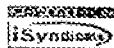
Fig. 4C

News and Research

Hello Direct: Telecom Services: Long Distance, Local, Wireless, and Enhanced

[Arizona Cities Airstrike Long Distance Slamming](#)
[The Days of Inphones Launches Hotcaller](#)
[Qwest Announces Free Internet Access for Consumers of Special Long Distance Pack](#)
[SBC Telecom Lowered Rates to Unprecedented Levels](#)

5:59 AM 06/26/1999
5:59 AM 06/26/1999
5:59 AM 06/26/1999
5:59 PM 06/19/1999
5:59 PM 06/15/1999



The BEST content for your web site!
Copyright 1999 iSyndicate, Inc.

Message Boards

Search
IP Address
Email

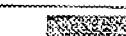
Ask Jeeves Community

Announcement
Welcome to Black Voices Discussion Forum! Feel free to change this message by clicking the edit profile button to your left.
Enjoy your discussions!!

The Cisco Systems ATM Interface Processor (AIP) for the Cisco 7000 family of router platforms, was the industry's first commercially available, standards-based Asynchronous Transfer Mode (ATM) router interface. (The Cisco 7000 family includes the Cisco 7200 and Cisco 7507 series routers.)

Search

Search For:



- Search the Following Sources:
- entire Web
 Research Centers Sites
 All Jeeves Sites
 Only Screens
 Only Videos
 Only News & Features
 Only Applications
 Only VC Home
 All Business Sites
 Only Associations
 Only Events
 Only Financial Firms
 Only News & Issues

**Small
Excite, Business**

Fig. 4D-1

By
Headline News
• Most SOHOs... v. HealthCare
Not Business Insurance
(8/20/99)

• Small Businesses Take On
More Credit (8/20/99)

Research Your Industry
Articles and abstracts from
Thousands of publications

[Select an industry] 

Business Owner's Handbook
The essential guide to running
your business.

Free Business Forms
Free business templates,
checklists, and forms.

Axera Demia v1.0, Copyright © 1999 Axera, Inc. All Rights Reserved.

Fig. 4D - 2

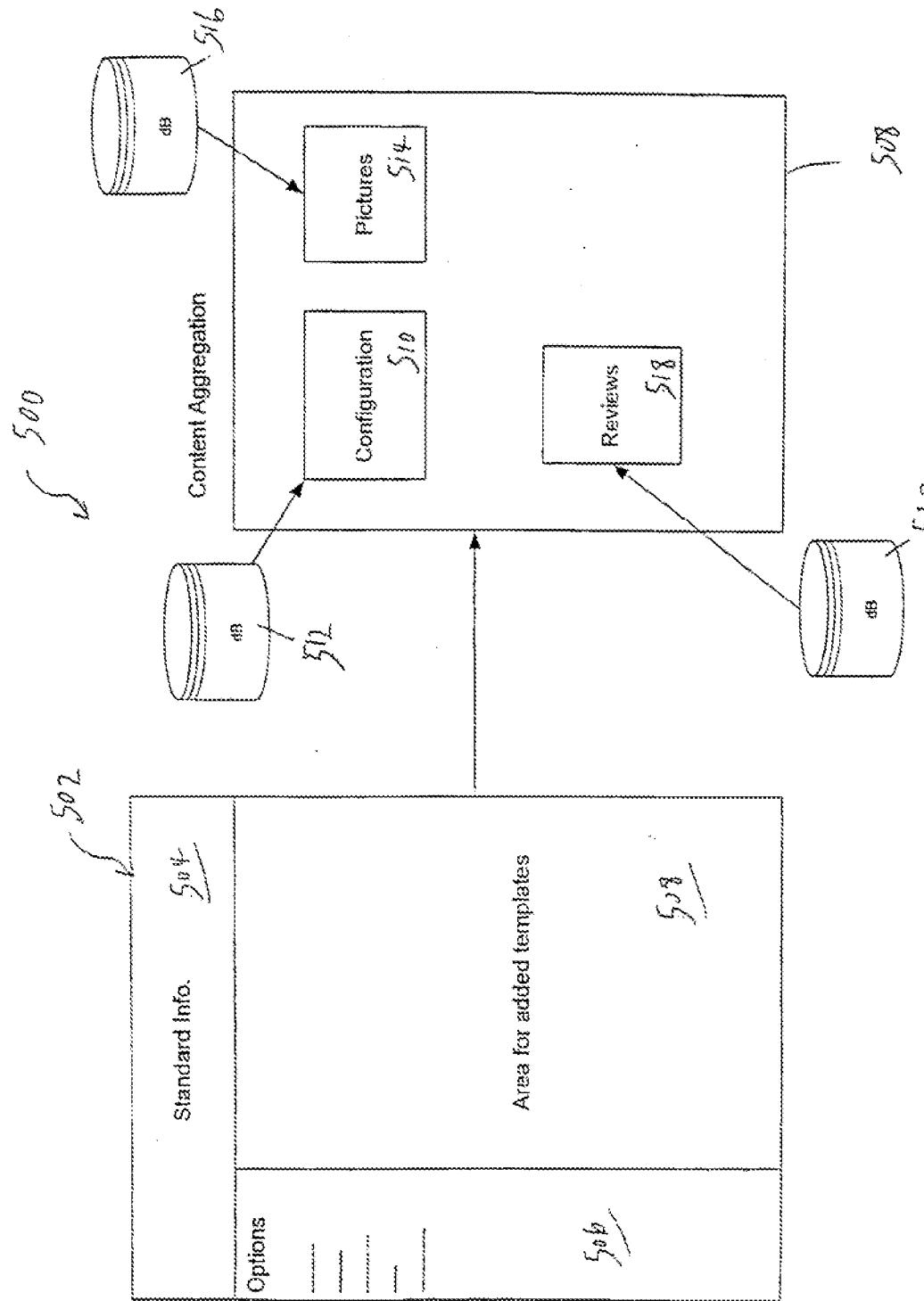


Figure 5

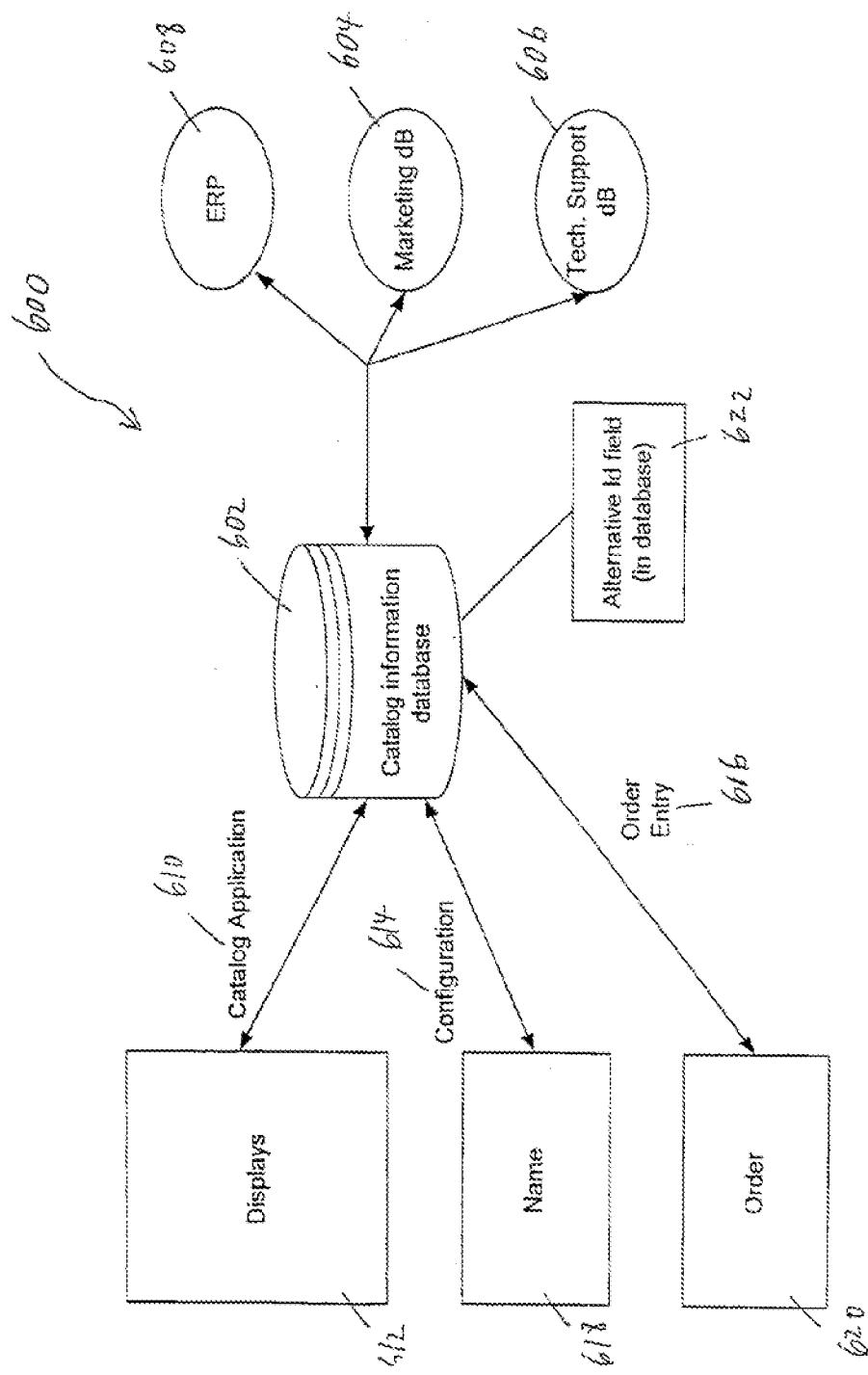


Figure 6

Served by Asera®
User Management

Catalog Search

Browse Catalog
Search Catalog

Search on Catalog Id or Alternate Id

Product Id:

Product List

Order Product	Product Id	Product Description
Add to Bente	FPC	Flexible PIC Concentrator
Cartonage	M40BASIC-AC	M40 Base Unit with AC Power
Add to Bente	SVC-COM-L	Service Core (1-4)
Add to Bente	SVC-COM-H	Service Core (5-8)
Add to Bente	SVC-ND-L	Service Next Day (1-4)
Add to Bente	SVC-ND-H	Service Next Day Service (5-8)
Add to Bente	SVC-SD-L	Same Day Service (1-4)
Add to Bente	DOC-M40-CD-S	Full Documentation Set, CD ROM
Add to Bente	DOC-M40-HW-S	Installation Guide, HW (hardcopy), AXI 520
Add to Bente	DOC-M40-HWS	Installation Guide, Hardware (hardcopy)

Page 1 of 9

Product Datasheet
Datasheets
Datasheet

Asera Demo v1.0 Copyright © 1999 Asera, Inc. All Rights Reserved.

an Asera service
Asera Demo v1.0 Copyright © 1999 Asera, Inc. All Rights Reserved.

Fig. 6A

Product Catalog Configuration Order Management Order Status User Management Served by Asera

Product Configuration

List Configurations

Product Name: M40BASE-DC

The following is a list of configured line items:

Configuration Name	Configuration ID	None								
Untitled										
Modify	Delete	Product Name								
		Configured M40								
		Line Item Details								
		<table border="1"> <thead> <tr> <th>Power Supply</th> <th>DC Power Supply</th> </tr> </thead> <tbody> <tr> <td>Slot 0/Port 0</td> <td>P-4DS3</td> </tr> <tr> <td>Slot 0/Port 1</td> <td>P-1GE-SX</td> </tr> <tr> <td>Slot 1/Port 0</td> <td>P-3OC3-RIM-SKIR</td> </tr> </tbody> </table>	Power Supply	DC Power Supply	Slot 0/Port 0	P-4DS3	Slot 0/Port 1	P-1GE-SX	Slot 1/Port 0	P-3OC3-RIM-SKIR
Power Supply	DC Power Supply									
Slot 0/Port 0	P-4DS3									
Slot 0/Port 1	P-1GE-SX									
Slot 1/Port 0	P-3OC3-RIM-SKIR									

Asera Demo v1.0. Copyright © 1999 Asera, Inc. All Rights Reserved.
an Asera service

SEARCH	RESULTS	SEARCHES
Served by Asera		
Product Catalog Configurator Order Status User Management		

Quote #: 000140

Ship To	A10001 AseraAccB1	Ship To	A10002 AseraAccS1	Quote Name
Bill To Address	300 Saginaw	Ship To Address	500 Tanber Road	Project Name
State	State 3	Building	Building 203	Created Date
Sales Forecast?	<input type="checkbox"/>	Manufacturing Forecast?	<input type="checkbox"/>	Booking probability [25 %]
Target Back Date	6/9/23/1999	Target Ship Date	6/10/23/1999	
View Details Print Email Delete Quote				
View Type Brief View Detailed View Quoted				

Line Item	Description	Quantity	Request Arrival Date	Unit Price	Discount Rate (%)	Extended Price	Delete
1	M000000000 M40 Base Unit with DC Power	1		\$62,000.00	0	\$182,000.00	<input type="checkbox"/>

Item List

#	Product ID	Description	QTY	Request Arrival Date	Unit Price	Discount Rate (%)	Extended Price	Delete
1	M000000000	M40 Base Unit with DC Power	1		\$62,000.00	0	\$182,000.00	<input type="checkbox"/>

Asera Demo v1.0, Copyright © 1999 Asera, Inc. All Rights Reserved.
an Asera service

Fig 66

14/37

Customer: [REDACTED]
Order Start Date: [REDACTED] **Order End Date:** [REDACTED]

Product Catalog: Configurator **Order Status:** [REDACTED]

Served by Asera: [REDACTED] **User Management:** [REDACTED]

Quote #: 000140

Configurable Items

Product ID	M40BASE-DC
Description	M40 Base Unit with DC Power
QTY	1
Unit Price	\$62,088.00
Discount Rate (%)	[REDACTED]
Extended Price	\$62,088.00
Request Arrival Date	

Config Items per Unit

#	Product ID	Description	Qty	Unit price	Discount Rate (%)	Extended price
1	PWR-M40-DC-S	Power Supply Spare, DC	1	\$16,200.00	[REDACTED]	\$16,200.00
2	P-4-DS3	PIG, DS3, 4 Port with cables	1	\$42,500.00	[REDACTED]	\$42,500.00
3	P-1-GR-SX	Ethernet PIC, 1000 Base-SX, 1 Port	1	\$30,000.00	[REDACTED]	\$30,000.00
4	P-2OC3-ATM-SMR	PIG, STM1/OC3 ATM SMF IR, 2 Port	1	\$32,000.00	[REDACTED]	\$32,000.00

Asera Demo v1.0 Copyright © 1999 Asera, Inc. All Rights Reserved.
An Asera service

Fig. 6D

15/37

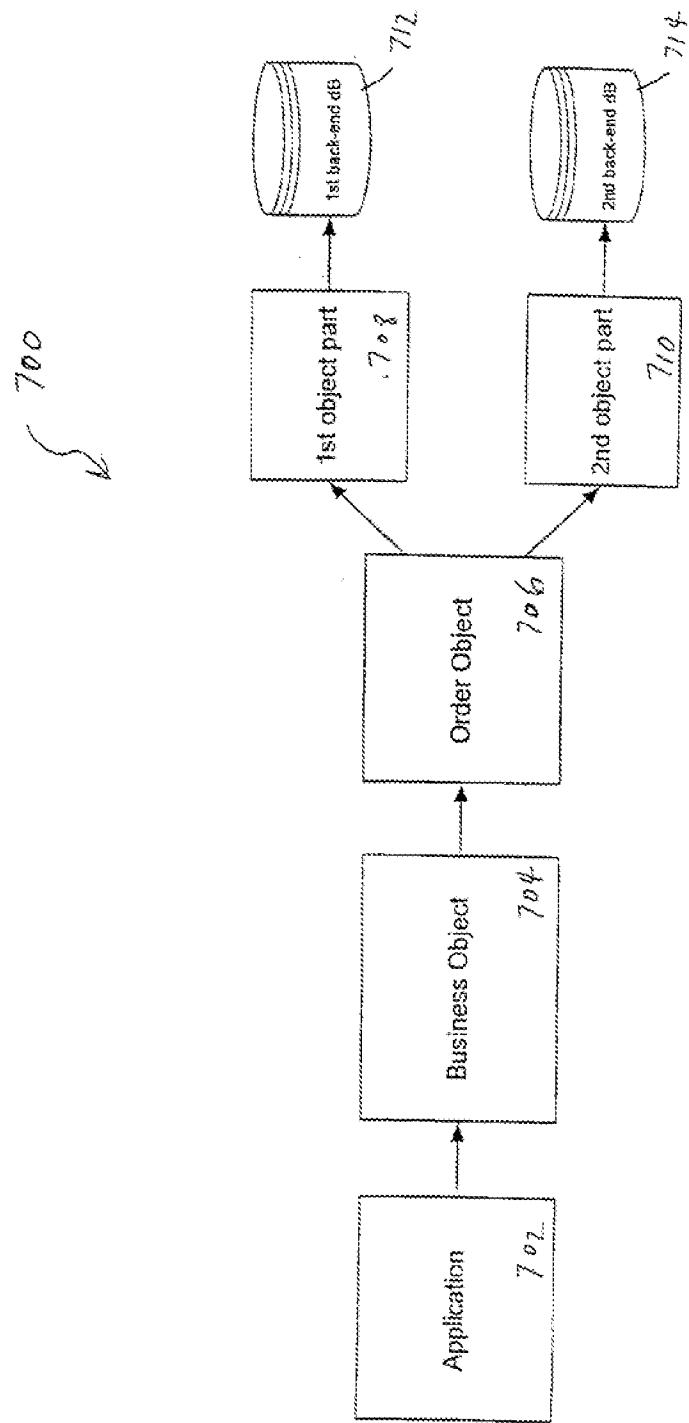


Figure 7

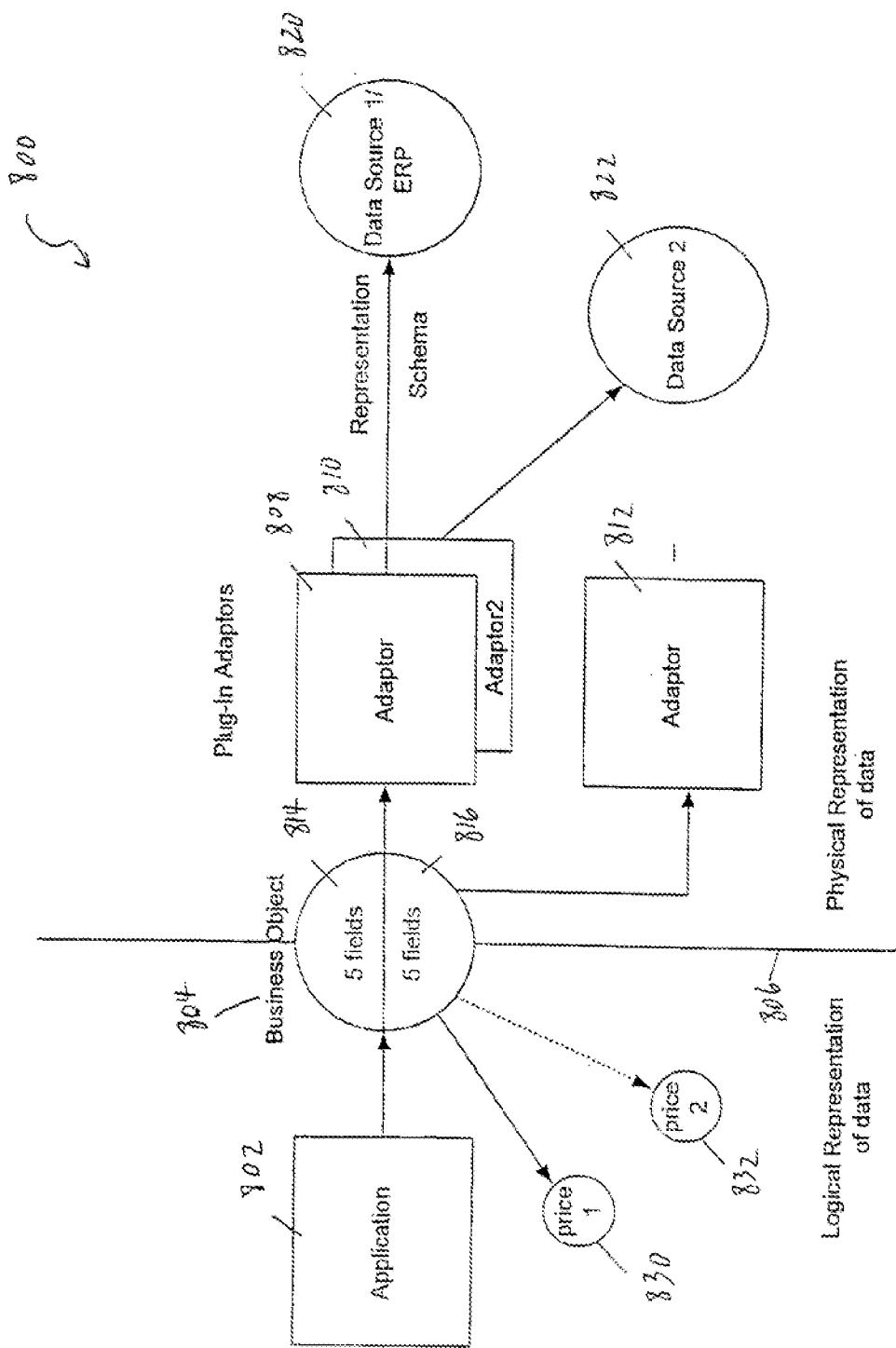


Figure 7

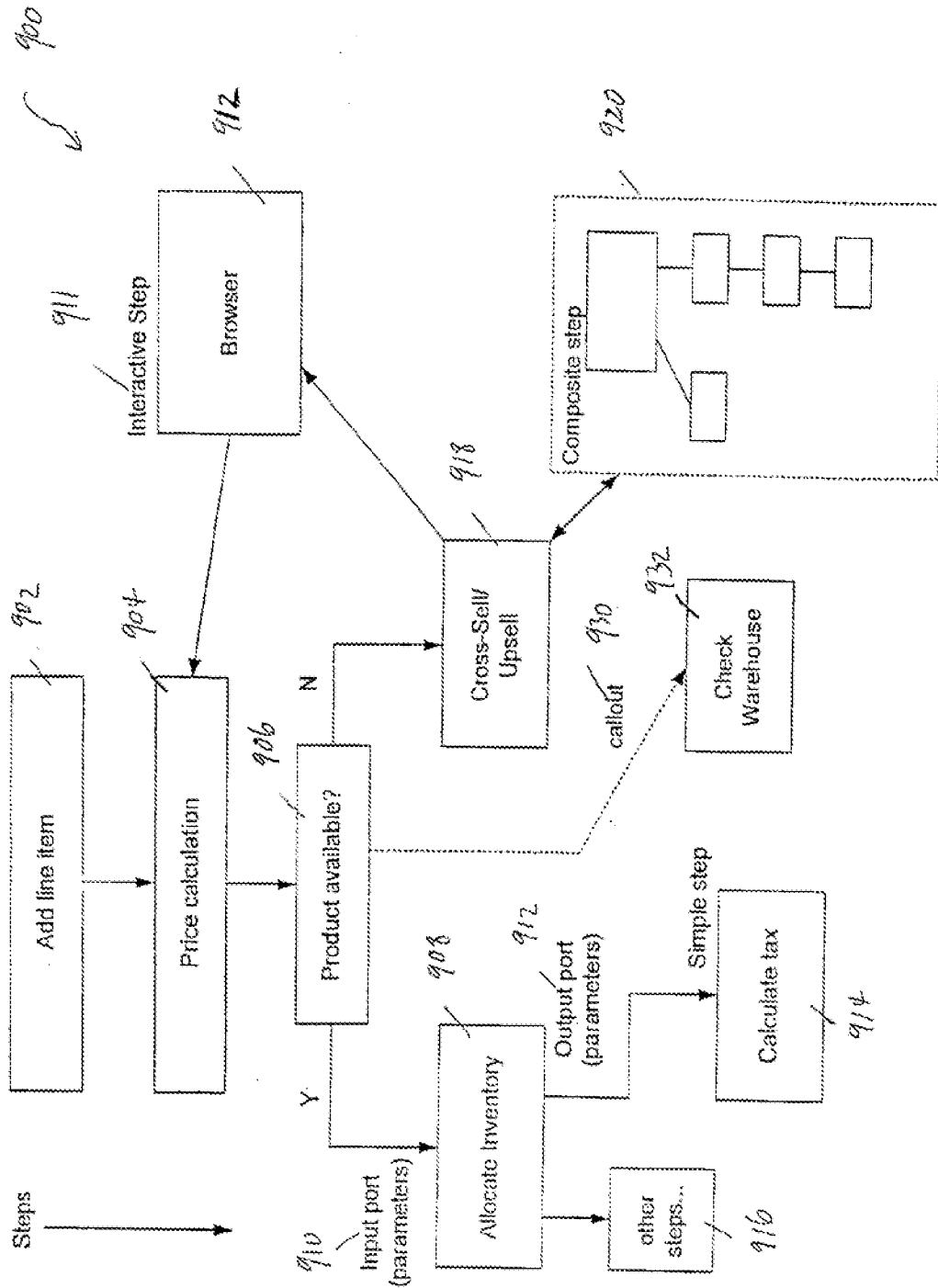


Figure 9

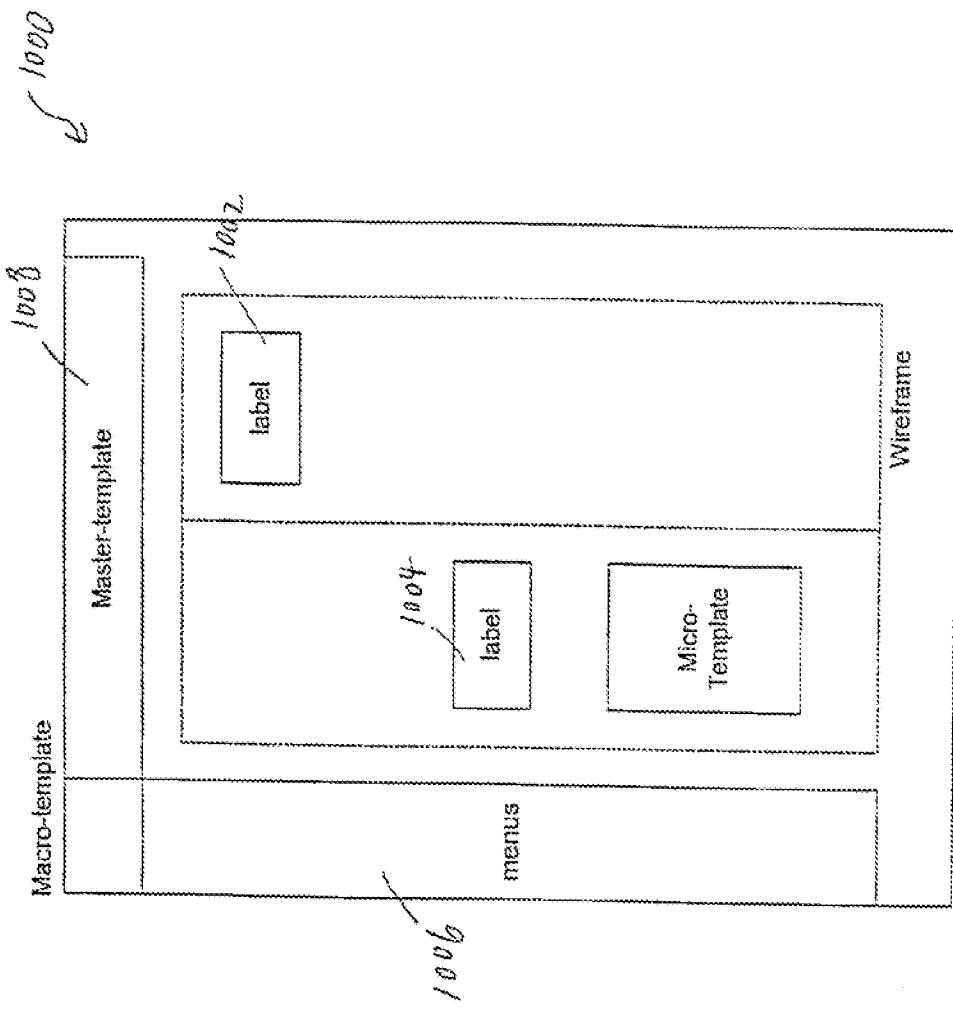


Figure 1D

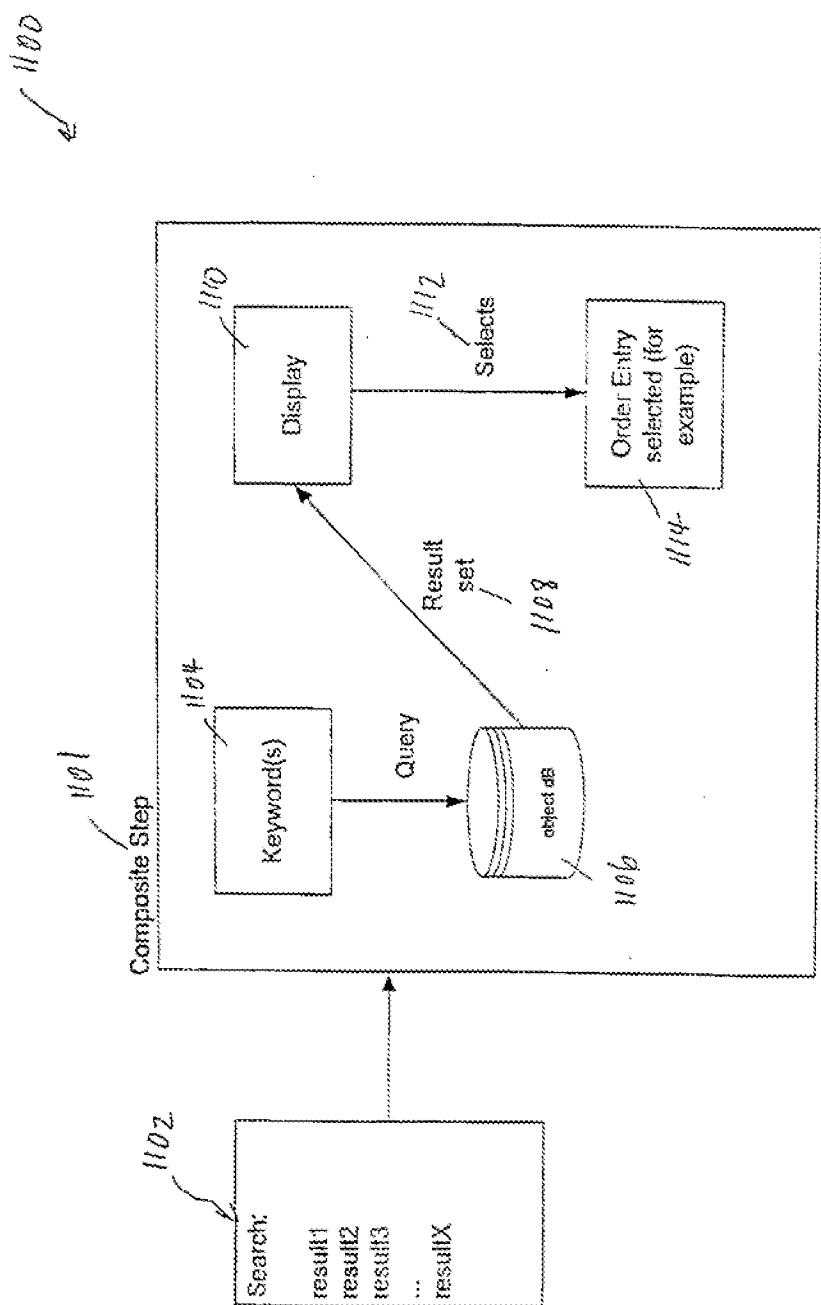


Figure 11

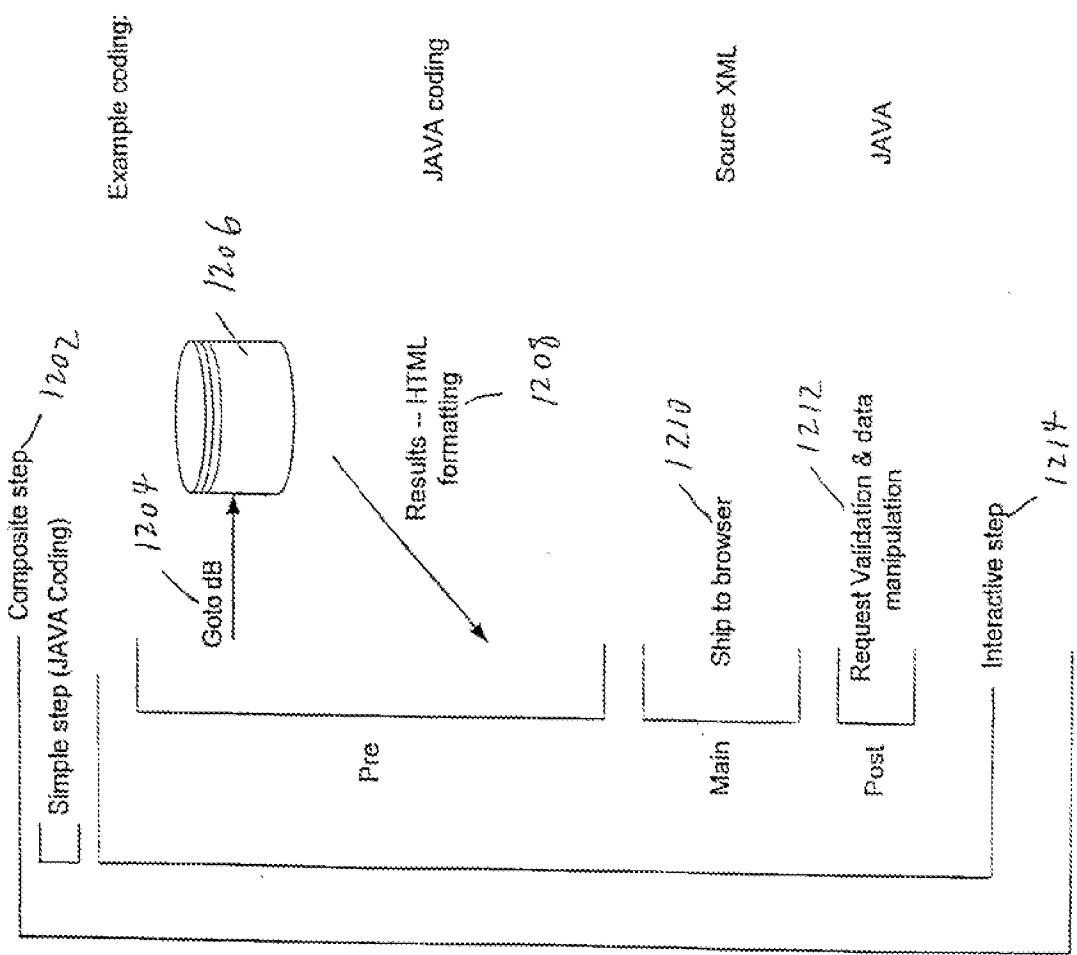


Figure 12

21/37

✓ 1300

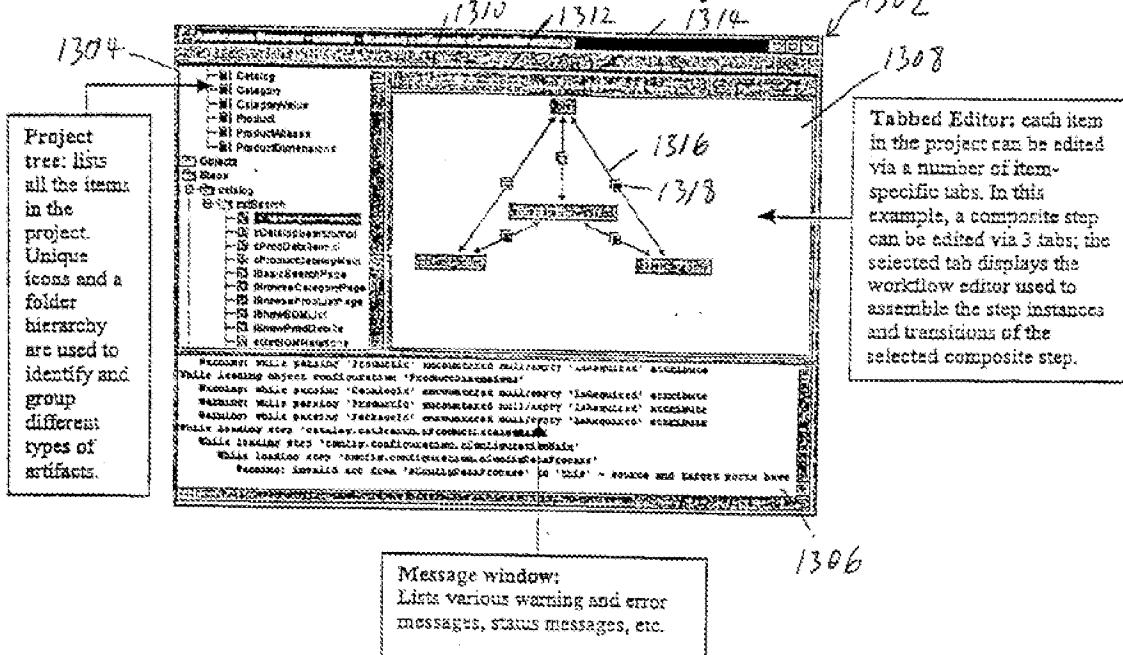
The Application Designer Window Anatomy

Fig. 13

1324

1322

1325

Application/DeclarativeCatalog - [Product]

```

Configurations
Objects
catalog
  SORMRelations
  Catalog
  Category
  CategoryRelation
  CategoryValue
  CompetitiveRef
  Packages
  Product
  ProductAnnames
  ProductCategory
  ProductDimension
  ProductDocument
  ProductSubstitution
  ProductSupersession
  SubstitutionType
  SupersessionRef

1324
1320

// This method retrieves a product object given a unique product key.
// catalog id are not supplied, current default catalog group is used
public static Product getProductByIds(Context context, String catalogId,
String productId) throws Exception {
    // first find out if it is in the cache
    String key = catalogId + ":" + catalogGroup + ":" + productId;
    BaseBusObj obj = (BaseBusObj)BusinessObjectPool.get(context, "Product");
    if (obj != null)
        return (Product)obj;

    // object is not found, go and load the object
    AccessOpInputDoc accInput = new AccessOpInputDoc();
    CallContext callCtx = new CallContext();

    callCtx.setAdapterName("CatalogAdapter");
    accInput.addCondition("", "CatalogID", AccessOpInputDoc.EQ, catalogId);
    accInput.addCondition("AND", "CatalogGroup", AccessOpInputDoc.EQ, catalogGroup);
}

Warning: while parsing 'ProductId' encountered null/empty 'IsRequired' attribute
Warning: while parsing 'PackageId' encountered null/empty 'IsRequired' attribute
While loading step 'catalog.catSearch.cProductCatalogMain'
  While loading step 'config.configuration.cConfigurationMain'
    While loading step 'config.configuration.cConfigDataProcess'
      Warning: invalid arc from 'cConfigDataProcess' to 'this' - source and target parts have
Internal error: object 'order.Quote' was added multiple times
While loading object configuration 'Quote'
  Warning: named filter configuration item 'ONQuoteList:ActiveQuots' is not tagged 'InUse' in de

```

Fig. 13A

1326

Application Designer - catalog - [Product]

Implementation Attributes

Name	Type	Cardinality	Required	Access	Operations
CatalogId	String	ISINGLE	Yes	Yes	Yes
BCMS_Cat	Integer	ISINGLE	Yes	No	Yes
BCMS_Ref	String	ISINGLE	No	No	Yes
BCMS_Cat	String	ISINGLE	No	No	Yes
Products	String	ISINGLE	Yes	Yes	Yes
ProductName	String	ISINGLE	Yes	No	Yes
SupplierPrd	String	ISINGLE	Yes	Yes	Yes
ProductDesc	String	ISINGLE	No	No	Yes

Relationships / Access Operations

Relationship	Access
catalog.catalogMain	Access
catalog.products	Access
catalog.productName	Access
catalog.supplierPrd	Access
catalog.productDesc	Access

Warnings:

- Warning: while parsing 'Products' encountered null/empty 'IsRequired' attribute
- Warning: while parsing 'PackageId' encountered null/empty 'IsRequired' attribute
- While loading step 'catalog.catalogMain'
- While loading step 'config.configuration.cConfigurationMain'
- While loading step 'config.configuration.cConfigDataProcess'
- Warning: invalid arc from 'cConfigDataProcess' to 'this' - source and target ports have
- Internal error: object 'order.Quota' was added multiple times
- While loading object configuration 'Quota'
- Warning: named filter configuration item 'CMQuotaList;ActiveQuota' is not tagged 'InUse' in da

Fig. 13B

24/37

1328

Relationship Descriptions - [Product]

The screenshot shows a software interface for managing object relationships. On the left, a tree view lists various objects and their relationships, with the 'catalog' object expanded to show its sub-relationships. The 'Product' object is highlighted. The right side of the interface contains a large table with columns for 'Relationship', 'Description', 'IsRequired', and 'Containment'. Below the table, a scrollable text area displays several warning messages related to object parsing and configuration.

Relationship	Description	IsRequired	Containment	
ProductAlias... ProductAlias...	MULTIPLE	No	Yes	Containment
SHOMMaterials BOMRelat...	MULTIPLE	No	Yes	Containment
ProductDim... ProductDim...	MULTIPLE	No	Yes	Containment
ProductDoc... ProductDoc...	MULTIPLE	No	Yes	Containment
ProductSubs... ProductSubs...	MULTIPLE	No	Yes	Containment
ProductSubs... ProductSups...	MULTIPLE	No	Yes	Containment

```

Warning: while parsing 'ProductID' encountered null/empty 'IsRequired' attribute
Warning: while parsing 'PackageIDs' encountered null/empty 'IsRequired' attribute
While loading step 'catalog.catSearch.cProductCatalogMain'
  While loading step 'config.configuration.cConfigurationMain'
    While loading step 'config.configuration.cConfigDataProcess'
      Warning: invalid arc from 'cConfigDataProcess' to 'this' - source and target ports have
      Internal error: object 'order.Quote' was added multiple times
      While loading object configuration 'Quote'
      Warning: named filter configuration item 'CMQuotelist:ActiveQuotes' is not tagged 'InUse' in de
  
```

Fig. 13C

25/37

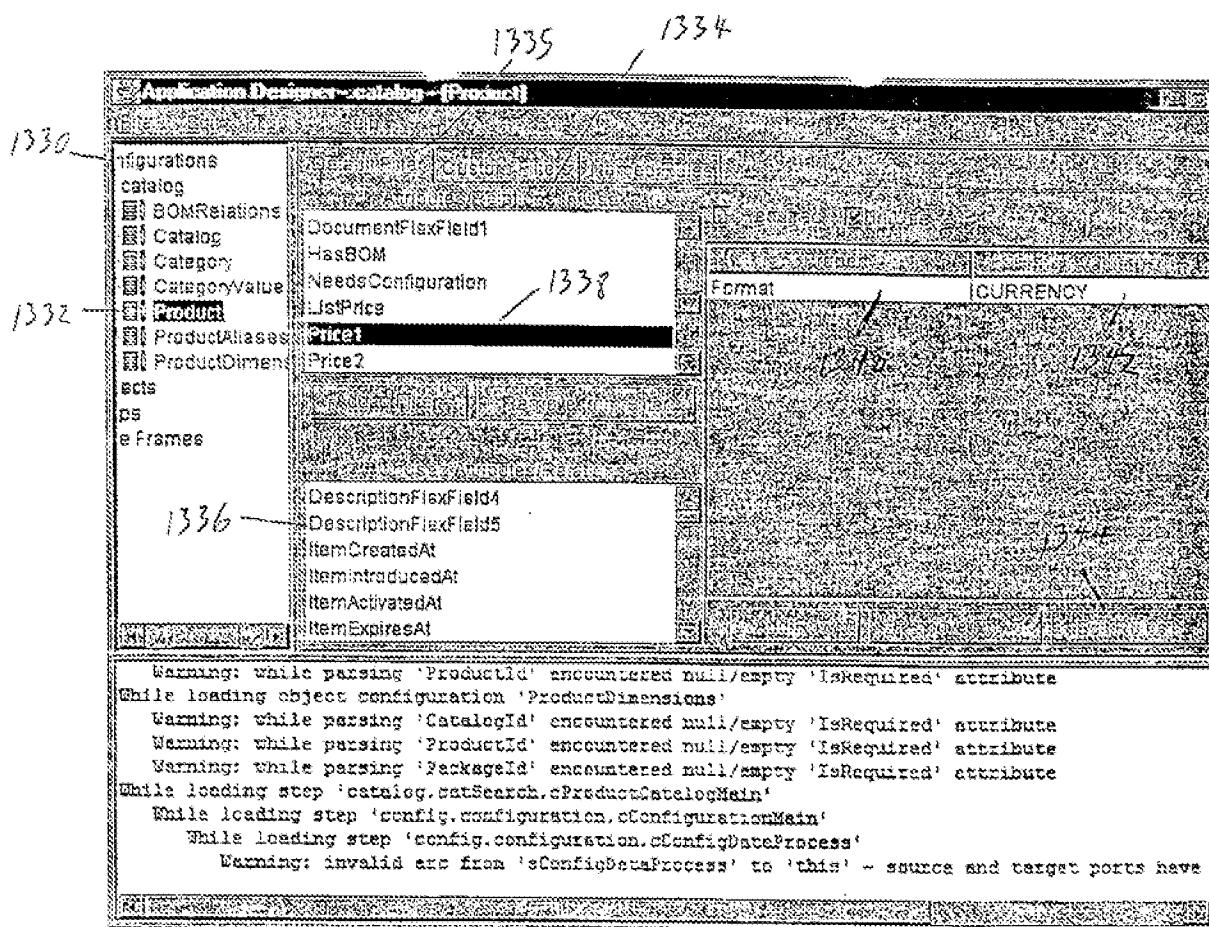


Fig. 130

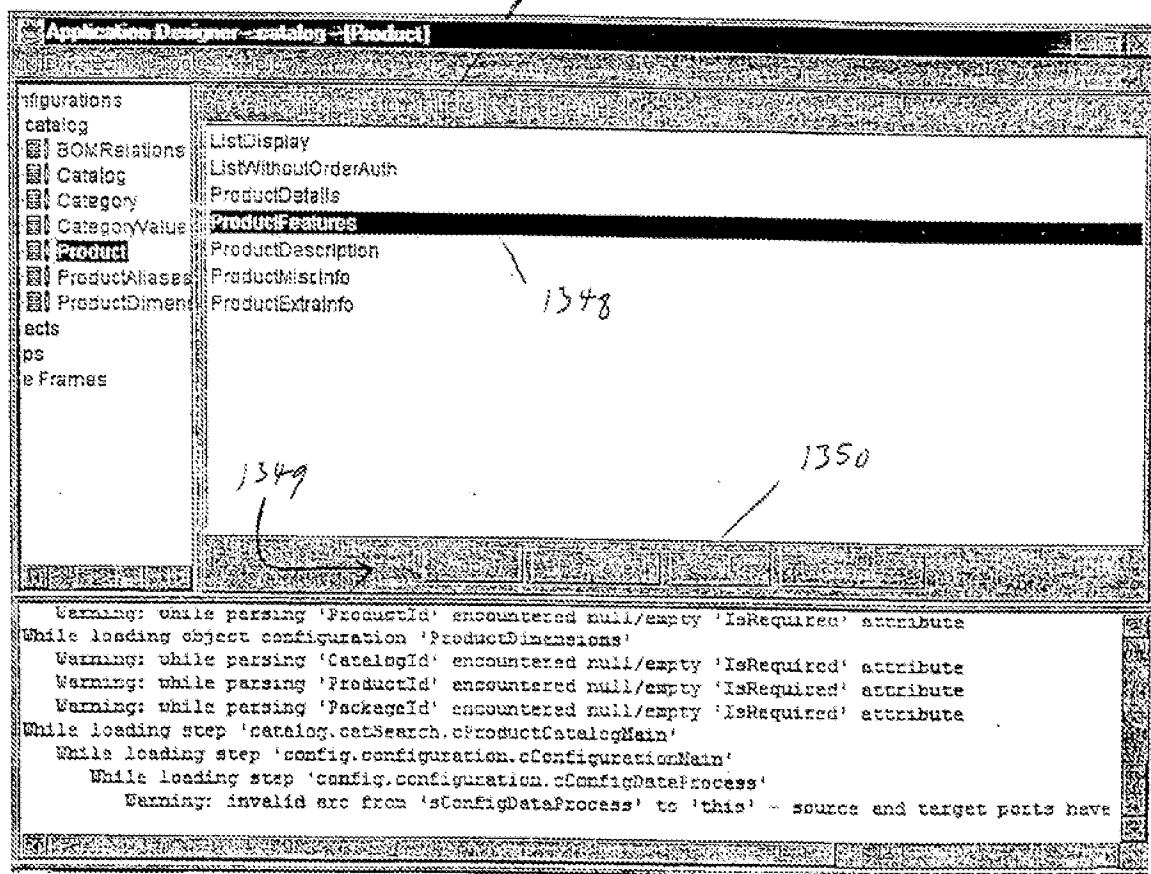


Fig. 13E

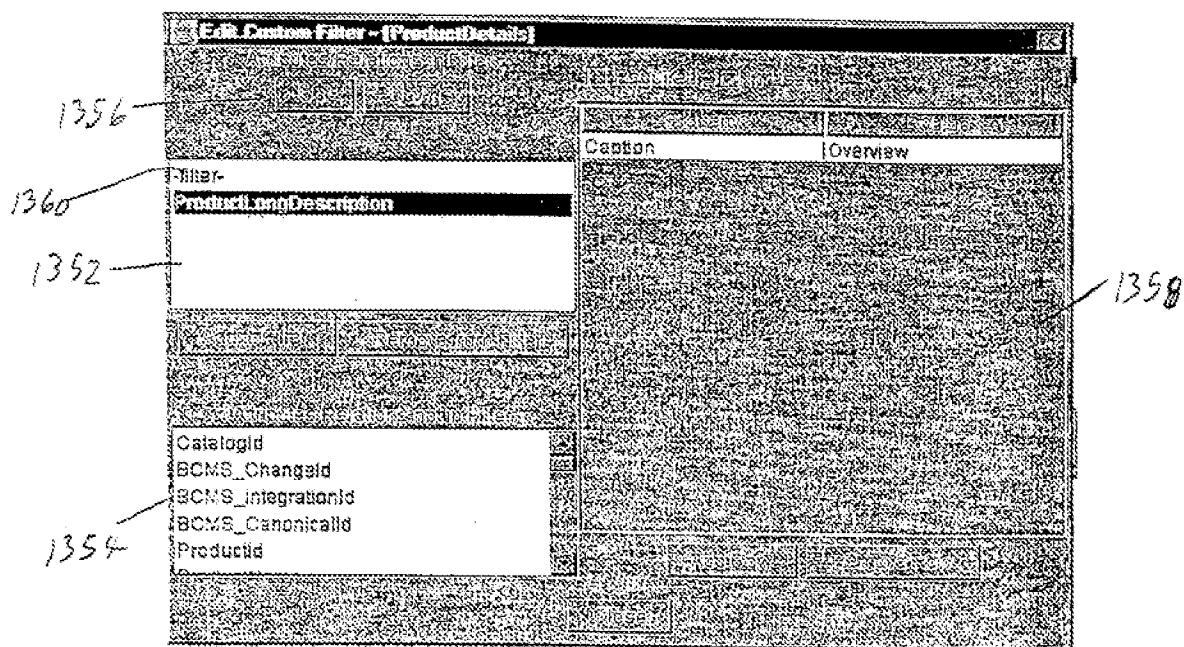


Fig. 13F

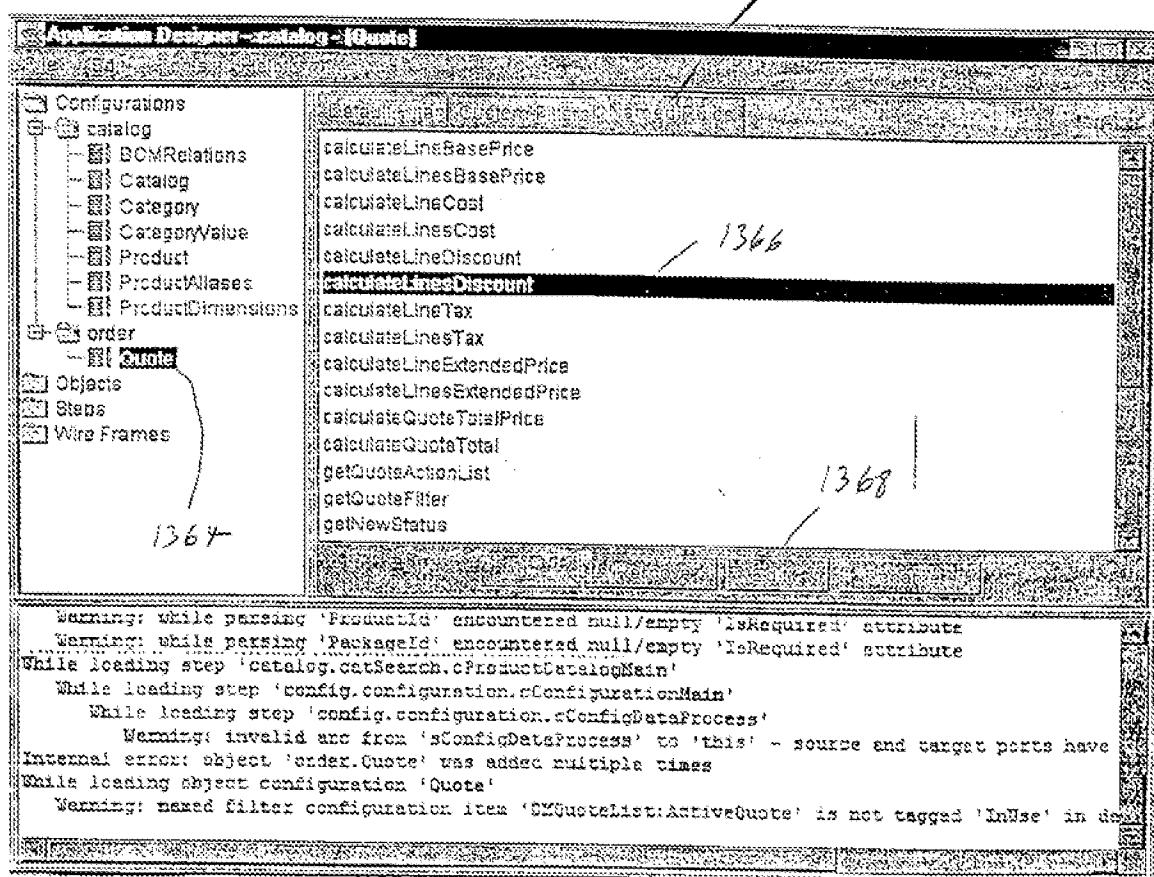


Fig. 13G

The screenshot shows a software window titled "Edit Item Rule - {calculateLinesDiscount}." The main area contains the following XML code:

```
<Action>
  <Method>
    <Class><![CDATA(boms.app.order.OrderEntryUtil.Discount]]></Class>
    <Method><![CDATA[calculateLinesDiscount]]></Method>
  </Method>
</Action>
```

Below the code, the number "1370" is displayed. At the bottom of the window, there is a toolbar with several icons.

Fig. 13H

1376

1378

Application Designer - catalog - [catalogSearchStepGetBOMRelations]

```

1372
1374

Configurations
Objects
Steps
catalog
  catSearch
    cCatalogBrowseImpl
    cCatalogSearchImpl
    cProdDetailImpl
    cProductCatalogMain
    iBasicSearchPage
    iBrowseCategoryPage
    iBrowseProdListPage
    iShowBOMList
    iShowProdDetails
    sGetBOMRelations
    sGetProductDetails
    sInitializeSearch
    sPrepareBrowsePage
    sPrepareSearchPage
config

String productId = (String)pInputParamList.findElement("ProductId");
AccessOpInputDoc ecclInput = new AccessOpInputDoc();
CallContext callCtx = new CallContext();
callCtx.setAdapterName("CatalogAdapter");

String inputPort = (pContext.getBusProcess()).getCurrentInputPort();
if (inputPort.equals("GetBOMListPort"))

{
  // add search criteria
  ecclInput.addCondition(aiOp, "ProductId", AccessOpInputDoc.OP_EQ);

  // Selection list
  ecclInput.addResultType(AccessOpInputDoc.BASE_ATTRIBUTES);

  // execute query
  BaseDocObj resObj = AccessMgr.execute(pContext, "Product");
  BusObjIterator resIterator = null;
  if (resObj != null)
    {
      Warning: while parsing 'ProductId' encountered null/empty 'IsRequired' attribute
      While loading object configuration 'ProductDimensions'
      Warning: while parsing 'CatalogId' encountered null/empty 'IsRequired' attribute
      Warning: while parsing 'ProductId' encountered null/empty 'IsRequired' attribute
      Warning: while parsing 'PackageId' encountered null/empty 'IsRequired' attribute
      While loading step 'catalog.catSearch.cProductCatalogMain'
        While loading step 'config.configuration.cConfigurationMain'
          While loading step 'config.configuration.cConfigDataProcess'
            Warning: invalid arc from 'cConfigDataProcess' to 'this' - source and target ports have
}

```

Fig. 13T

31/37

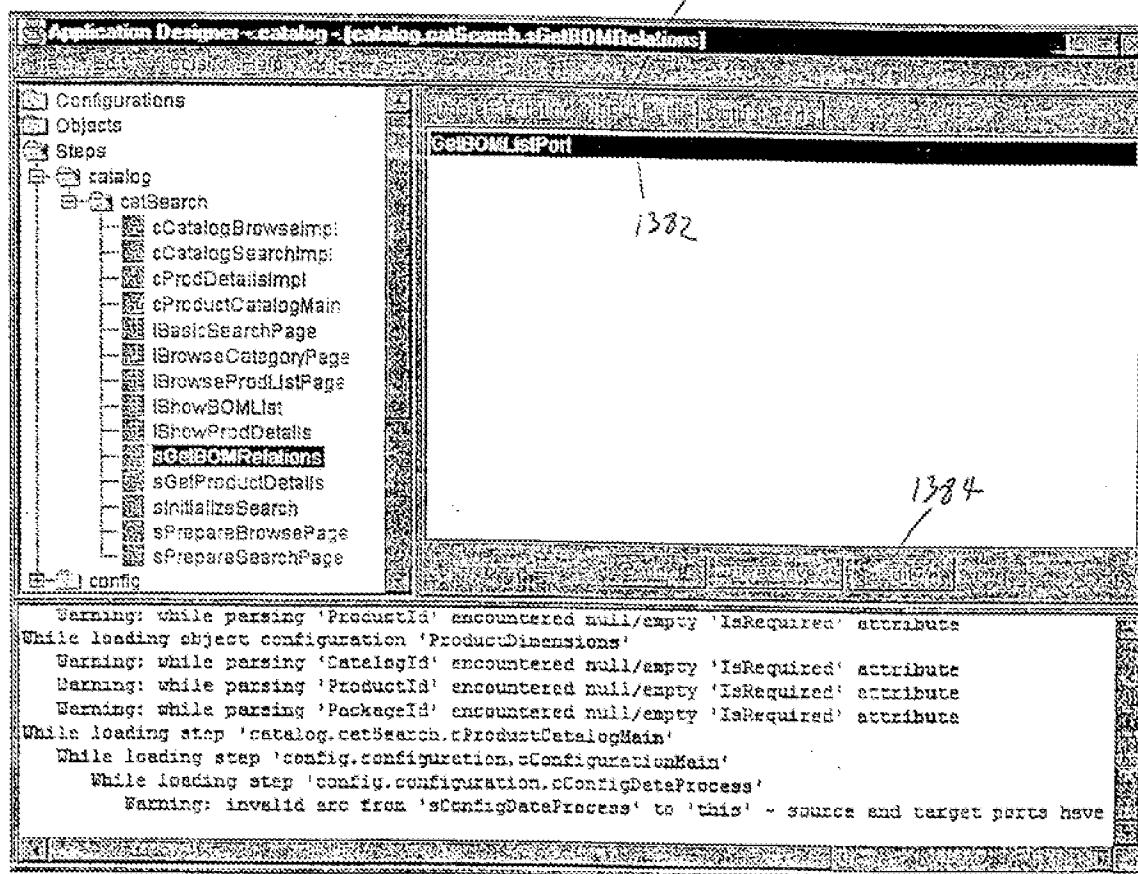


Fig. 13J

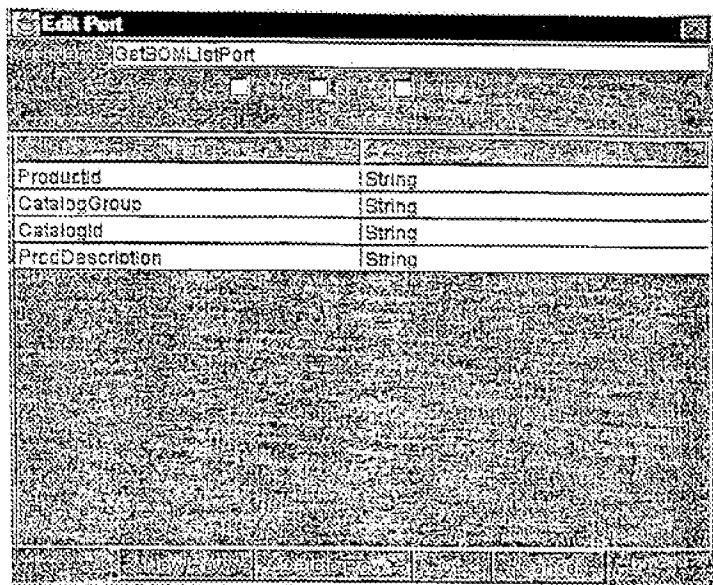
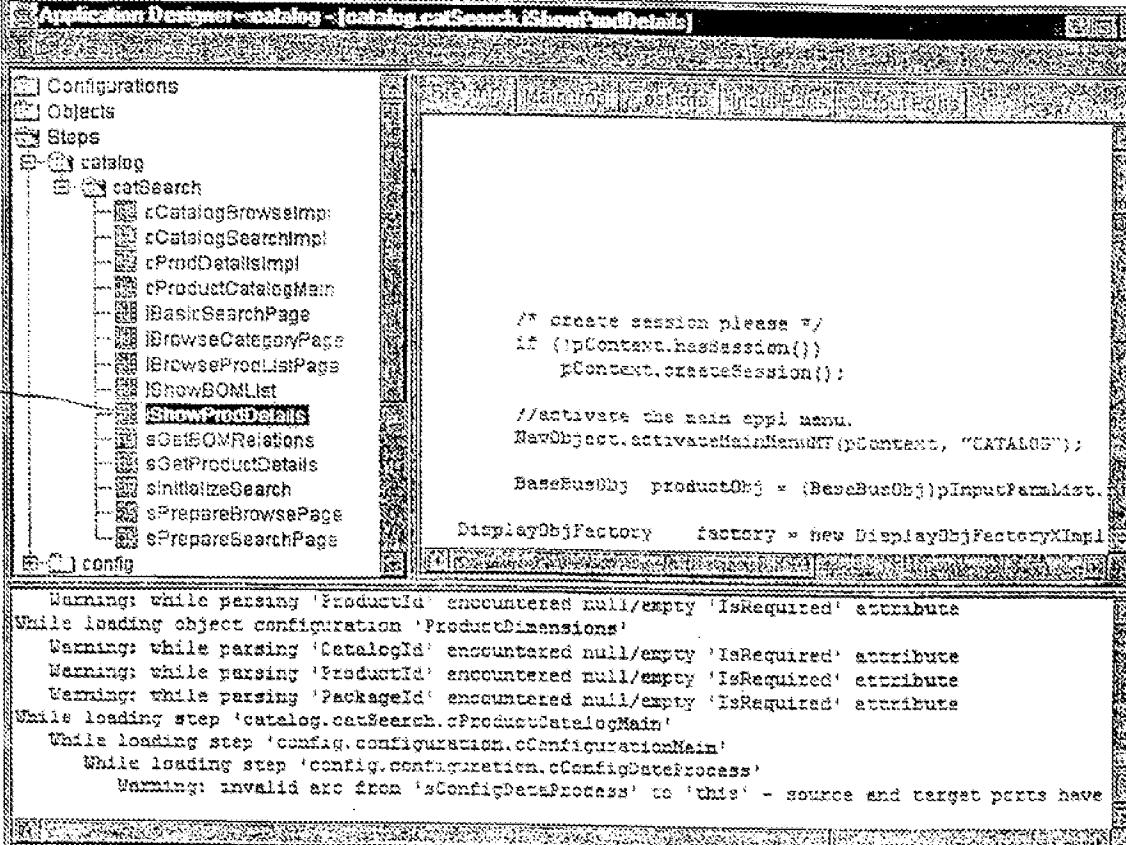


Fig. 13K

138



The screenshot shows the Application Designer interface with the title bar "Application Designer - catalog - [catalog.catSearch.ShowProdDetails]". The left pane displays a tree view of configurations, objects, steps, and a catalog category containing a catSearch step. The right pane shows the code for the ShowProdDetails step:

```

/*
 * create session please
 */
if (!pContext.hasSession())
    pContext.createSession();

//activate the main appi menu.
NewObject.activateMainMenuItem(pContext, "CATALOG");

BaseBusObj productObj = (BaseBusObj)pInputParamList.
DisplayObjFactory factory = new DisplayObjFactoryImpl();

```

Below the code, a warning message is displayed:

```

Warning: while parsing 'ProductId' encountered null/empty 'IsRequired' attribute
While loading object configuration 'ProductDimensions'
Warning: while parsing 'CatalogId' encountered null/empty 'IsRequired' attribute
Warning: while parsing 'ProductId' encountered null/empty 'IsRequired' attribute
Warning: while parsing 'PackageId' encountered null/empty 'IsRequired' attribute
While loading step 'catalog.catSearch.cProductCatalogMain'
    While loading step 'config.configuration.cConfigurationMain'
        While loading step 'config.configuration.cConfigDataProcess'
            Warning: invalid arc from 'cConfigDataProcess' to 'this' - source and target ports have

```

1386

Fig. 13L

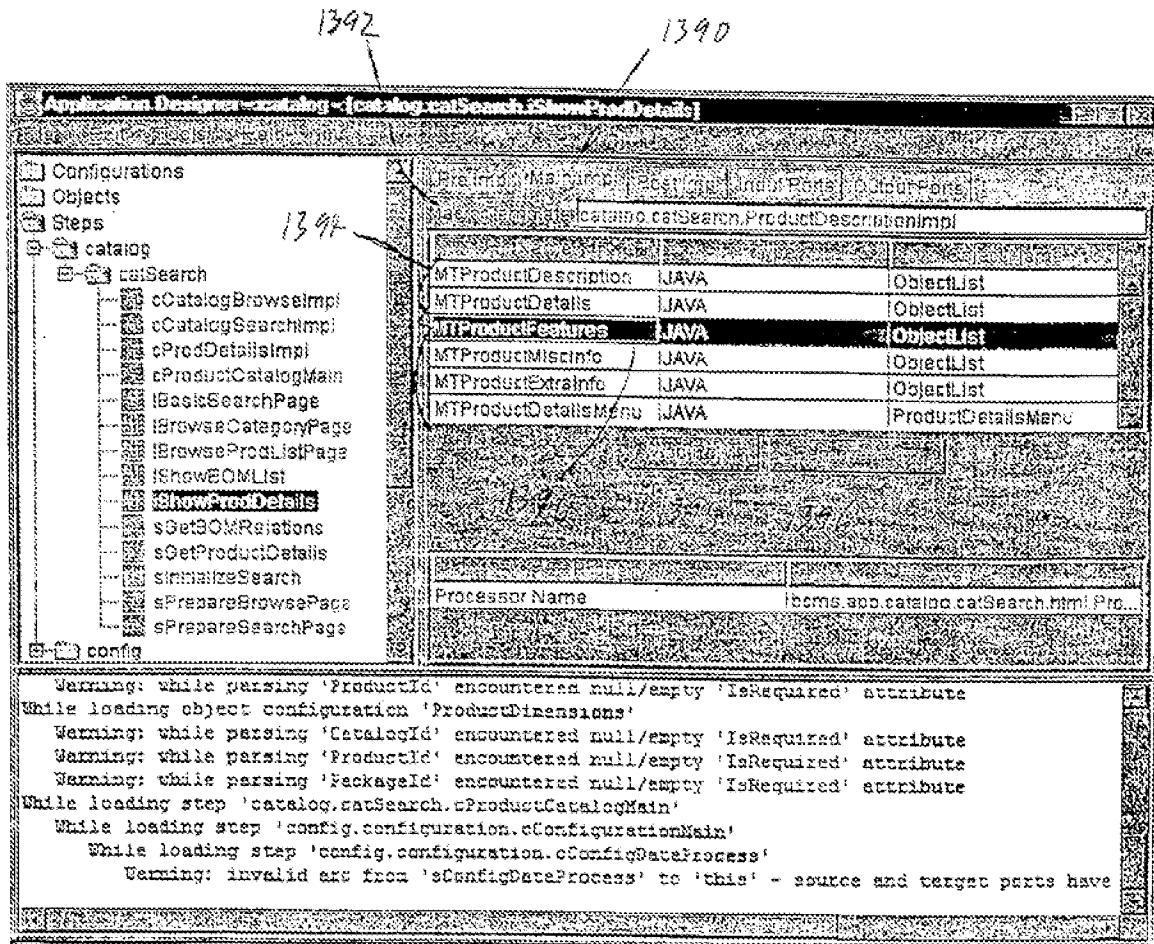


Fig. 13M

1398

The screenshot shows the Application Designer interface with the title bar "Application Designer - catalog - (catalog.catSearch.Step.ProductDetails)". The left pane displays a tree view of configurations, objects, steps, and a catalog category containing several sub-steps. The right pane shows the step details for "catalog.catSearch". The code editor contains Java-like pseudocode for handling different output ports based on the current bus process. A warning message at the bottom of the code editor lists various parsing errors and loading warnings.

```

String outputPort = (pContent.getBusProcess()), getCurrentOutputPort();
String tmpStr;

if (outputPort.equals("OrderProductPort"))
{
    tmpStr = (String)pInputParamList.findElement("OrderQty");
    Integer qty = new Integer(tmpStr);

    BaseDocObj tmpDoc = new BaseDocObj("BCMSProduct");
    tmpDoc.setStrAttr("BCMSCatalogGroup", (String)pInputParamList.getStrAttr("BCMSCatalogID"), (String)pInputParamList.getStrAttr("BCMSPartID", (String)pInputParamList.getStrAttr("BCMSQuantity", qty.intValue()));

    BaseDocObj partsDoc = new BaseDocObj("PartsDoc");
    partsDoc.setProp("NonConfigProduct", "true");
}

Warning: while parsing 'ProductId' encountered null/empty 'IsRequired' attribute
while loading object configuration 'ProductDimensions'
Warning: while parsing 'CatalogId' encountered null/empty 'IsRequired' attribute
Warning: while parsing 'ProductId' encountered null/empty 'IsRequired' attribute
Warning: while parsing 'PackageId' encountered null/empty 'IsRequired' attribute
While loading step 'catalog.catSearch.cProductCatalogMain'
    While loading step 'config.configuration.cConfigurationMain'
        While loading step 'config.configuration.cConfigDataProcess'
            Warning: invalid arc from 'cConfigDataProcess' to 'this' - source and target ports have

```

Fig. 13n

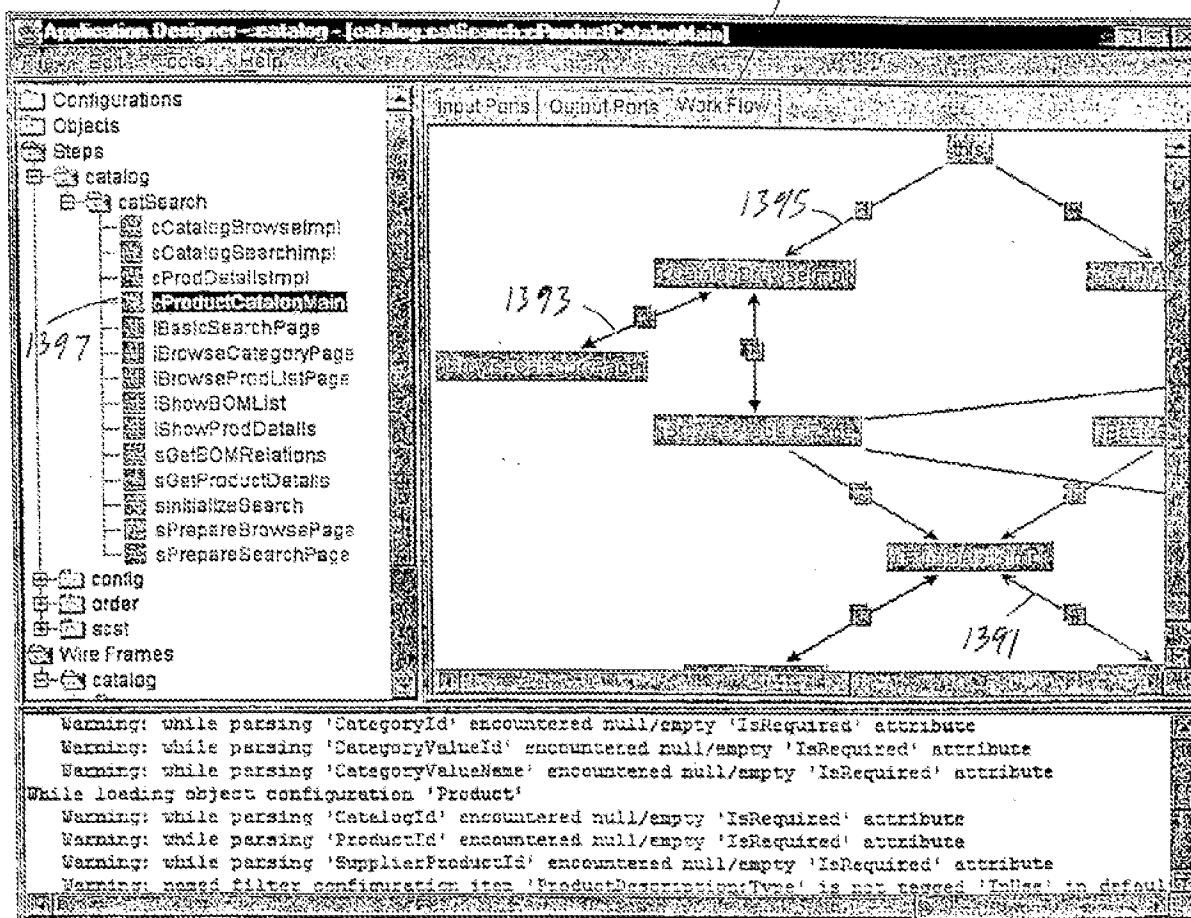


Fig. 130

INTERNATIONAL SEARCH REPORT

Intern'l Application No
PCT/US 00/22032

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 915 115 A (TALATI KIRIT K) 22 June 1999 (1999-06-22) column 2, line 12 -column 4, line 34 ---	1,27,40, 65,80, 93,99
A	US 5 875 330 A (GOTI JUAN CARLOS) 23 February 1999 (1999-02-23) column 1, line 17 -column 2, line 19 ---	1,27,40, 65,80, 93,99
A	US 5 668 978 A (YASUTAKE KOUICHI ET AL) 16 September 1997 (1997-09-16) column 1, line 12 -column 3, line 35 --- -/-	1,27,40, 65,80, 93,99

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *U* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

& document member of the same patent family

Date of the actual completion of the international search

Date of mailing of the international search report

1 December 2000

08/12/2000

Name and mailing address of the ISA

European Patent Office, P.O. 5818 Patentkantoor 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2046, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Brandt, J

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/22032

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GB 2 320 111 A (JBA HOLDINGS PLC) 10 June 1998 (1998-06-10) page 3, line 23 -page 10, line 13 -----	1, 27, 40, 65, 80, 93, 99
A	US 5 418 957 A (NARAYAN ROM) 23 May 1995 (1995-05-23) column 1, line 12 -column 7, line 52 -----	1, 27, 40, 65, 80, 93, 99

INTERNATIONAL SEARCH REPORT

Information on patent family members

Interr	1st Application No
PCT/US 00/22032	

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5915115	A	22-06-1999	US	5390330 A	14-02-1995
			AU	4759796 A	23-10-1996
			WO	9631828 A	10-10-1996
			US	5677997 A	14-10-1997
			US	5999942 A	07-12-1999
			AU	677835 B	08-05-1997
			AU	6392094 A	29-08-1994
			CA	2155865 A	18-08-1994
			EP	0686285 A	13-12-1995
			WO	9418629 A	18-08-1994
US 5875330	A	23-02-1999	US	5638539 A	10-06-1997
US 5668978	A	16-09-1997	JP	6149554 A	27-05-1994
			DE	69311065 D	03-07-1997
			DE	69311065 T	20-11-1997
			EP	0597646 A	18-05-1994
GB 2320111	A	10-06-1998	AU	5401498 A	29-06-1998
			CN	1245571 A	23-02-2000
			EP	0943126 A	22-09-1999
			WO	9825203 A	11-06-1998
US 5418957	A	23-05-1995	NONE		